1970

The numerical optimization of distributed parameter systems by gradient methods

Douglas Edward Cornick
Iowa State University

THE NUMERICAL OPTIMIZATION OF DISTRIBUTED PARAMETER SYSTEMS BY GRADIENT METHODS

by

Douglas Edward Cornick

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major Subjects: Aerospace Engineering
Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

Heads of Major Departments

Signature was redacted for privacy.

Dean of Graduate College

Iowa State University
Ames, Iowa

1970

TABLE OF CONTENTS

	Page
CHAPTER I. INTRODUCTION	1
Results Concerning the Problem Formulation	3
Results Concerning the Problem Solution	5
Engineering Applications	7
Dissertation Objectives	8
Class of Problems Considered	10
Dissertation Outline	11
CHAPTER II. THE OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS	14
The Optimal Control Problem	14
The Distributed Parameter Optimal Control Problem	16
Conditions for Optimality	19
Methods of Solution	24
CHAPTER III. INTRODUCTION TO GRADIENT METHODS	26
Gradient Method Algorithm	28
General Results for Gradient Methods	40
CHAPTER IV. AN APPROXIMATION THEORY FOR GRADIENT METHODS	42
The Effects of the Discrete Approximations on the Gradient Vector	46
The Effects of Gradient Error on Gradient Methods	55
Error Estimates	68
Determination of the Parameters in the Error Estimates	77

Page

Geometric Interpretation of the Error Bounds 80

CHAPTER V. NUMERICAL RESULTS 84

CHAPTER VI. CONCLUDING REMARKS 111

LITERATURE CITED 118

ACKNOWLEDGMENTS 124

APPENDIX A. MATHEMATICAL PRELIMINARIES 125

Selected Results from Functional Analysis 125

Pertinent Results from Partial Differential Equations 133

Pertinent Results from Approximation Theory and Numerical Analysis 136

APPENDIX B. THE NON-LINEAR DISTRIBUTED PARAMETER OPTIMAL CONTROL PROBLEM 141

Problem Formulation 141

Derivation of the Necessary Conditions 142

Optimality Conditions 146

CHAPTER I. INTRODUCTION

The optimal control of distributed parameter systems is concerned with the minimization (maximization) of functionals constrained by either nonhomogeneous partial differential equations or by multiple integral equations. The study of the optimal control of distributed parameter systems is generally considered to have been initiated in 1960 by Butkovskii and Lerner (1). The term "distributed parameter system" was coined by Butkovskii and was intended to refer to dynamical systems which are modeled by either partial differential equations or by multiple integral equations. In fact, it is these distributed constraints which distinguishes this field from the classical multi-variable calculus of variations, which was considered by Lagrange as early as 1806.

The optimal control of distributed parameter systems has received considerable attention in recent years. Since Butkovskii and Lerner's original paper, well over two hundred publications have appeared in the literature concerned with this subject. At the present time two full length books (2, 3) have been published on this topic and more are in preparation. In addition many of the recent texts on optimization include chapters introducing this subject (4, 5). Also, a number of doctoral dissertations have reported results on various aspects of this field (6, 7, 8 and 9).

The rapid growth of literature concerning this topic has

motivated a number of recent survey papers on this subject, which include extensive bibliographies. The first survey of the subject was generated by Wang (10) in 1964 and still provides a good introduction to the subject. Subsequently, Wang published an extensive bibliography covering both the stability and the optimal control of distributed parameter systems (11). In 1968 Butkovskii, Egorov and Lurie (12) published an excellent survey of the Soviet efforts in this field. In 1969, Robinson compiled what is probably the most complete bibliography of this subject to date (13). Robinson includes in his bibliography a brief discussion of many of the various facets of this topic. Due to the existence of these recent survey papers, only a brief introduction to distributed parameter systems will be given; and a complete bibliography will be omitted.

At the present time there is no universally accepted method for classifying the works published on this subject. A number of possibilities are discussed in (13). In the subsequent discussion, the publications will be divided into two groups: (1) those papers which are primarily concerned with the mathematical structure of the problem formulation; and (2) those results which are primarily concerned with the problem solution.

Results Concerning the Problem Formulation

The majority of the papers on the optimal control of distributed parameter systems deal primarily with the extensions of the theoretical results obtained for lumped parameter systems to distributed parameter systems. In fact about one-half of all the reports, which have appeared in the literature, are concerned with the problem formulation giving particular attention to the derivation of the necessary conditions for optimality. Three basic approaches have been utilized in the derivation of these necessary conditions: (1) variational calculus; (2) dynamic programming; and (3) functional analysis. In addition, there is the method of moments which can be used if the functional is constrained by a system of linear integral equations (14). In his early works, Butkovskii (15, 16) considers systems described by integral equations and employs variational methods to derive the necessary conditions for optimality. These necessary conditions are given in the form of integral equations. A number of Butkovskii's subsequent works are concerned with the development of methods for solving these integral equations. Egorov (17) and Lurie (18) follow the work of Butkovskii; however, they consider systems described by partial differential equations. Both of these approaches have their advantages and their disadvantages. Since the integral representation of the dynamical system yields bounded

operators, this approach is useful in theoretical developments. However, the differential representation of dynamical systems, which unfortunately introduces unbounded operators, is useful because many physical problems are easily formulated in terms of partial differential equations. In principle at least, Green's functions can be employed to convert linear partial differential equations into linear integral equations. However, in practice this is not always possible, certainly not in general for the non-linear case.

Wang was one of the first to use dynamic programming to derive the necessary conditions for distributed parameter systems with distributed control. Brogan (6) extends Wang's results to include boundary control. The functional analysis methods are generally applied to abstract optimal control problems in either a Banach space or in a Hilbert space. With this degree of generality, the results obtained in these papers certainly can be applied to distributed parameter systems and to lumped parameter control problems as well. Papers by Balakrishnan (19, 20) are of particular interest to the present investigation; since in these papers, Balakrishnan considers the extension of the classical steepest descent method to the general Hilbert space setting. Russell (21) applies functional analysis methods to problems in which the controls are finite-dimensional. Axelband (22) utilizes the Frećhet derivative to obtain the necessary conditions for a

quadratic functional. He then proceeds to develop methods for solving the resulting linear operator equation. A number of authors, for example (23, 24 and 25) utilize certain properties of special classes of problems to develop methods for obtaining the optimal control.

Results Concerning the Problem Solution

In the optimization of distributed parameter systems, one's first impulse is to transform the problem into some other form which can be solved by existing techniques. This approach leads ultimately to some type of approximation. At the present, the following approximations have been tried: (1) eigenvector expansion; (2) spacial discretization; and (3) space-time discretization. Of course, the eigenvector (eigenfunction is the term usually used in this case) expansion techniques are classical methods of approximating partial differential equations. Unfortunately, this method only works for linear or linearized problems with rather restrictive boundary conditions. When this method does apply, the distributed parameter problem is reduced to a lumped parameter problem. An approximation is introduced when the eigenvector expansion is truncated to a finite number of terms in order to facilitate a practical solution. Lukes and Russell (26) prove that the solution obtained from the truncated eigenvector expansion converges to the exact solu-

tion of the distributed parameter problem as the number of terms in the expansion increases. Space discretization also reduces the problem to an approximate lumped parameter problem. However, a very large number of ordinary differential equations result from this method; and conventional lumped parameter methods have not proven to be very effective in this case. Axelband (22) uses space-time discretization to reduce the problem to a parameter optimization problem. However, once again the number of independent variables becomes extremely large; and difficulties are encountered in obtaining the solution with standard techniques. Axelband also proves that in the limit this method converges to the true solution. However, in doing so, he neglects to consider the numerical approximations and their effects on the convergence of the method. Recently, a number of authors (27, 28, 29 and 30) have alluded to the fact that some of the direct computational methods developed for the solution of lumped parameter optimization problems, especially gradient methods, might also be beneficially extended to distributed parameter systems. At the present time computational experience with the steepest descent method, as related to the optimization of distributed parameter systems, has been reported in (28, 29 and 30). These methods offer the advantages of being very simple and of applying to a broad class of problems.

Engineering Applications

In recent years considerations of the control of complex processes, such as nuclear reactors and chemical production systems, have motivated interest in the optimal control of distributed parameter systems. However, these are not the only possible applications for this theory. For example, it is clear that an optimal control theory for distributed parameter systems can be applied to the process industries (chemical, petroleum, steel, cement, glass, etc.), the power industry, and the aerospace industry. The following list is not complete; nevertheless, it does indicate the variety of problem areas to which the optimal control theory of distributed parameter systems could be applied:

1. Control of heat and mass transfer processes (e.g., heating, cooling, melting, drying, etc.).

2. Control of fluid dynamic processes (e.g., pumping of petroleum, hydroelectric power generation, liquid rocket engine design, acoustic phenomena, etc.).

3. Control of chemical and kinetic reactions (e.g., petroleum refining, production of steel and glass, combustion processes, chemical industries, etc.).

4. Control of elastic and viscoelastic vibrations (e.g., heavy equipment industry, aerospace industry, geographic applications, location of petroleum deposits, etc.).

5. Control of nuclear and atomic processes (e.g., nuclear power industry, nuclear space propulsion systems, nuclear energy propagation, etc.).

6. Control of radioactive processes (e.g., radiation shielding, optical and electro-magnetic communications, etc.).

7. Control of hydrodynamical and magnetohydrodynamic processes.

8. Control of spacecraft attitude (e.g., heat dissipation, structural effects, etc.).

9. Control of melting, freezing, and crystal growth.

10. Control of environmental processes (e.g., air pollution, water pollution, flood control, traffic control, forest fire control, etc.).

After examining the above list, it becomes immediately apparent that there is no lack of motivation (from the point of view of applications) for the theory of optimal control of distributed parameter systems.

Dissertation Objectives

As mentioned before, many of the existing results to date are concerned with the mathematical structure of the problem and the derivation of the necessary conditions for optimality. Unfortunately, very little has been said concerning how to solve these necessary conditions to obtain the optimal control. From the engineering point of view, the problem solution is at least as important as the problem formulation. Therefore, it seems desirable that a large amount of future research efforts should be devoted to the development of methods for solving the problems already formulated.

One of the original objectives of the present research was to demonstrate numerically that the second generation

gradient methods, such as the conjugate gradient method and the Davidon method, could be efficiently adapted to solve practical distributed parameter problems. These methods were selected because of their simplicity, their generality, and their success in solving lumped parameter optimal control problems. However, preliminary storage requirement calculations indicate that the solution of realistic distributed parameter optimization problems are beyond the present storage capabilities of the Model 360-65 system. In addition, early numerical results indicate that the approximations involved in discretizing the continuous problem are causing substantial errors in the approximate solution. It was realized that in order to effectively solve distributed optimal control problems by gradient methods, it is essential to determine the effects of these approximations on the numerical solution. The new objectives formulated are: (1) to develop a general optimization theory for a particular class of distributed parameter problems; (2) to isolate those approximations which cause the largest errors in the numerical solution for this class of problems; (3) to determine the effects of these errors on the class of gradient methods; (4) to develop estimates for the errors between the exact and the approximate solution; (5) to evaluate the effectiveness of the conjugate gradient method and the Davidon method in comparison with the standard steepest descent method on this class of

problems; and (6) to generate numerical results which substantiate the theory developed in objectives (1) through (5).

Class of Problems Considered

The non-linear distributed parameter optimal control problem is easily formulated; and if the existence of a relative minimum is assumed, the derivation of the necessary conditions for optimality is straight forward. However, the solution of a non-linear distributed parameter optimal control problem is usually very difficult. Existence and uniqueness considerations for both the minimizing element and the distributed dynamical system dictate that extreme care be exercised in the selection of the class of problems to be considered.

Fortunately, gradient methods do not require the a priori assumption of the existence of a relative minimum. However, they do require the existence and uniqueness of the solutions of the dynamical system. Thus, the selection of the distributed dynamical system to be optimized is an important consideration in distributed parameter problems.

The second generation gradient methods are basically unconstrained, quadratic functional, optimization methods. Thus, it seems natural to investigate their performance on quadratic distributed parameter problems, especially, since quadratic problems play such a significant role in the present state of

the art of distributed parameter systems. The penalty function approach can be used to alter the constrained distributed parameter optimal control problem into an unconstrained, quadratic functional, optimization problem; if: (1) the penalty functional is quadratic; (2) the original cost index is quadratic; and (3) the distributed parameter dynamical system is linear. In the following, only problems with the above properties will be considered; and will be referred to as quadratic programming problems.

Dissertation Outline

The distributed parameter optimal control problem is formulated in Chapter II. The concept of a functional derivative is utilized to derive the expression for the gradient of the cost index. Brief remarks are made concerning methods which use the gradient to obtain the optimal control.

Gradient methods are introduced in Chapter III. Specifically, an introduction of the three most popular gradient methods is presented. The concepts of the inner and outer loop iterations are discussed, and popular inner loop iterators are introduced.

The development of an approximation theory for the numerical solution of distributed parameter systems by gradient methods is presented in Chapter IV. The definitions of the Optimal Control Error and the Cost Functional Error

are introduced. It is shown that the approximations involved in the discretization of the continuous problem cause gradient errors. The effects of gradient error on gradient methods is analyzed. Error estimates for the approximate numerical solution are developed. A geometrical interpretation of these error estimates is presented.

Chapter V presents numerical results for both the constrained and the unconstrained optimal control of the one-dimensional wave equation. Both distributed control and boundary control are considered. Penalty functions are used to render the constrained problem amenable to the gradient methods. Standard numerical comparisons between the conjugate gradient method, the Davidon method, and the steepest descent method are given. Some of the numerical considerations, such as selection of appropriate finite-difference methods, multiple quadrature formulas, storage requirements, computer run times, etc., are discussed

Concluding remarks and recommendations for additional research are given in Chapter VI.

Appendix A introduces mathematical concepts which are pertinent to this dissertation. The coverage of these topics is extremely brief; consequently, it is not intended to be an introduction to any of the areas discussed, but rather as a point of reference for the development presented in the main body of this work.

In Appendix B the derivation of the necessary conditions for a general non-linear distributed parameter optimal control problem is presented. Ordinary differential equations on the spatial boundary are considered. The standard calculus of variations is employed in the derivation of the necessary conditions for optimality.

CHAPTER II. THE OPTIMAL CONTROL OF DISTRIBUTED PARAMETER SYSTEMS

The Optimal Control Problem

The optimal control problem may be stated as follows: minimize

$$J[u;x], \tag{2.1}$$

subject to

$$\psi[u;x] \geq \Theta, \quad x\varepsilon X \text{ and } u\varepsilon U; \tag{2.2}$$

where X is called the state space, U the set of admissible controls, and J[u;x] is a real valued functional defined on the product space U x X. The non-linear operator ψ is defined on U x X, and Θ is the null vector of this product space. The functional $J[\cdot;\cdot]$ is generally referred to as the cost index, and the conditions of Equation 2.2 are called the constraints. As a consequence of the constraints, the state trajectory x(t) is dependent upon the control u. Thus any particular optimal control problem depends on the nature of the functions J and ψ, and on the sets X and U. Consider the following special cases: 1. Parameter Optimization: let X and U be real Euclidean vector spaces, let ψ be a vector valued function, and let J be a scalar valued function; 2. Lumped Parameter Optimization: let X and U be properly selected function spaces, let ψ be decomposed into two operators T and S, where T represents an algebraic equality

and/or inequality constraint, and where S denotes a differential and/or integral operator with respect to one variable, and let J be a real valued functional; 3. Distributed Parameter Optimization: let X and U be properly selected function spaces, let ψ be composed of algebraic, differential and/or integral operators with respect to more than one variable, and let J be a real valued functional. The solution of problems formulated in case 3 (above) is the topic of this dissertation.

The difficulties encountered in solving for the optimal control of a distributed parameter system are generally related to the complexity of the constraints, Equation 2.2. For distributed parameter systems very few general results are available concerning the existence and uniqueness of the solution to the constraint equation. Consequently, little can be said regarding the solution of the general distributed parameter problem. However, there exist certain classes of problems, of practical significance, for which results can be obtained. For one such important class one lets: (1) $J[u;x]$ be a quadratic functional in u and x; and (2) $\psi[u;x]$ be a linear equality constraint. It is this particular class of distributed parameter problems which will be considered in what follows.

The Distributed Parameter Optimal Control Problem

Cost index

Let $J[u;x]$ be a real valued quadratic functional defined on the real separable Hilbert space $U \times X$, generated by the self-adjoint operators M and N, and by the inner product $\langle \cdot,\cdot \rangle$, and let $J[u;x]$ be specified by

$$J[u;x] = c_0 + \langle c_1,u\rangle + \frac{1}{2}\langle u,Mu\rangle + \langle c_2,x\rangle + \frac{1}{2}\langle x,Nx\rangle, \tag{2.3}$$

or

$$J\left[\begin{bmatrix}u_d\\u_b\end{bmatrix};x\right] = c_0 + \left\langle \begin{bmatrix}c_1^d\\c_1^b\end{bmatrix}, \begin{bmatrix}u_d\\u_b\end{bmatrix}\right\rangle + \frac{1}{2}\left\langle \begin{bmatrix}u_d\\u_b\end{bmatrix}, [M_d|M_b] \begin{bmatrix}u_d\\u_b\end{bmatrix}\right\rangle + \langle c_2,x\rangle + \frac{1}{2}\langle x,Nx\rangle, \tag{2.4}$$

where $X \subset L^2[\Omega x T]$, $U \subset L^2[\Omega x T]$, $\Omega \subset R^m$, $T \subset R^1$, $c_0 \varepsilon R^1$, $c_1 \varepsilon U$, $c_2 \varepsilon X$, and where the vector c_1 and the operator M are partitioned into $\begin{bmatrix}c_1^d\\c_1^b\end{bmatrix}$ and $[M_d|M_b]$, respectively. At any time $t \varepsilon T$ the distributed state of the system is denoted by

$$x(r,t) = \begin{bmatrix} x_1(r_1,r_2,\ldots,r_m,t) \\ \vdots \\ x_n(r_1,r_2,\ldots,r_m,t) \end{bmatrix} , \tag{2.5}$$

where $r\varepsilon\Omega$, and where each component $x_i(r_1,r_2,\ldots,r_m,t)\varepsilon X$, $i=1,2,\ldots,n$. Let the control vector u denote both the distributed control and the boundary control, that is

$$u = \begin{bmatrix} u_d(r,t) \\ \vdots \\ u_b(r_b,t) \end{bmatrix} . \tag{2.6}$$

The distributed control vector u_d is represented by

$$u_d(r,t) = \begin{bmatrix} u_d^1(r_1,r_2,\ldots,r_m,t) \\ \vdots \\ u_d^p(r_1,r_2,\ldots,r_m,t) \end{bmatrix} , \tag{2.7}$$

where each component $u_d^i(r_1,r_2,\ldots,r_m,t)\varepsilon U$, $i=1,2,\ldots,p\leq n$. The boundary control vector u_b is represented by

$$u_b(r_b,t) = \begin{bmatrix} u_b^1(r_b^1,r_b^2,\ldots,r_b^m,t) \\ \vdots \\ u_b^k(r_b^1,r_b^2,\ldots,r_b^m,t) \end{bmatrix} , \tag{2.8}$$

where $r_b\varepsilon\partial\Omega$, and each component $u_b^i(r_b^1,r_b^2,\ldots,r_b^m,t)\varepsilon U$, $i=1,2,\ldots,k\leq n$.

Constraints

Let the constraint $\psi[u;x]$ be decomposed into the linear distributed dynamical system

$$Sx(r,t) = u_d(r,t) \tag{2.9}$$

with initial condition

$$x(r,0) = x_0(r), \tag{2.10}$$

and terminal condition (target set)

$$\Psi_1 x(r,T_f) = x_D(r), \tag{2.11}$$

and into the boundary condition

$$Tx(r_b,t) = u_b(r_b,t), \tag{2.12}$$

where S and T are linear differential operators consisting of a linear combination of a time differential operator D_t, and a spatial differential operator D_r, given by

$$S \equiv c_1 D_t + c_2 D_r , \tag{2.13}$$

$$T \equiv [c_3 D_t + c_4 D_r]\Big|_{\partial\Omega} , \quad c_i \varepsilon R^1, \tag{2.14}$$

and Ψ_1 is a nxn self-adjoint matrix, and $x_D(r)$ is the desired state of the final time. In addition it is assumed that Equations 2.9, 2.10, 2.11, and 2.12 satisfy the conditions of Theorem 1.1 in (20), which insures the well-posedness of the dynamical system, and the representation of the solution in terms of integral operators. The distributed optimal control problem for the system of Equations 2.9, 2.10, 2.11, and 2.12 with respect to the cost index J, and the set of admissible controls U can now be restated as follows:

determine the control $u^* \varepsilon U$ such that

$$J[u^*;x(u^*)] = \min_{u \varepsilon U}\{J[u;x(u)]\} \quad . \tag{2.15}$$

Conditions for Optimality

Existence of an optimum

For the class of problems considered the existence and the uniqueness of the minimizing element u^* can be easily proven (31). This, of course, is certainly not the case for the general distributed parameter optimal control problem, since existence and uniqueness results for even the dynamical system do not (in general) exist. The existence and uniqueness of the solution was one of the primary reasons for the selection of this particular class of problems, as the subject of the present investigation.

Derivation of the necessary conditions for optimality

The numerical methods which are used in this dissertation are directly applicable to only the unconstrained problem. Thus, it is convenient to transform this constrained problem into some equivalent unconstrained problem.

Assume that the distributed dynamical system defined by Equations 2.9 through 2.12 satisfy the conditions of Theorem 1.1 (20); this insures that the dynamical system has a unique solution for all $u \varepsilon U$. However, only the controls in a subset

$U_\psi \subset U$ drive the system from the initial state to the target set. Therefore, the specification of a target set causes the dynamical system to generate a constraint in the control space. Hence, strictly speaking only the controls $u \varepsilon U_\psi$ are admissible, since if $u \varepsilon [U_\psi]^c$ the u does not satisfy all of the constraints.

The penalty function method will be employed to render the constrained problem amenable to gradient methods. The original problem is then replaced by an equivalent unconstrained problem. The only requirement of a penalty function is that it be a positive measure of the constraint violation. Thus for any particular problem, the penalty function is not unique. In the subsequent development the following penalty function will be utilized:

$$P[x(r,T_f)] = \langle \Psi, W\Psi \rangle = \langle \Psi_1 x(r,T_f) - x_D(r), W(\Psi_1 x(r,T_f) - x_D(r)) \rangle \tag{2.16}$$

where W is a nxn self-adjoint matrix of penalty constants.

The constrained problem may be restated as an unconstrained problem as follows:

minimize

$$J_p[u;x] = J[u;x] + P[x(r,T_F)] \tag{2.17}$$

subject to Equations 2.9, 2.10, and 2.12 (Note: Equation 2.11 is omitted). In the unconstrained problem the dynamical system is not a constraint, but rather a side condition, which

must be solved only to evaluate the penalized cost index J_p.

By representing the solution of the state system in terms of appropriate Green's functions it is possible to remove the explicit dependence of the cost index on the state trajectory. Thus, let the solution of the dynamical system exist, be unique, and have the representation

$$x(r,t) = \Phi(t)x_0(r) + S^{-1}(t)u_d(r,t) + T^{-1}(t)u_b(r_b,t), \quad (2.18)$$

where $\Phi(t)x_0(r)$ denotes the contribution to the solution at time t due to the initial conditions, $S^{-1}(t)u_d(r,t)$ denotes the contribution to the solution at time t due to the distributed control, and $T^{-1}(t)u_b(r_b,t)$ denotes the contribution to the solution at time t due to the boundary control. The state of the system at the time T_f is then denoted by

$$x(r,T_f) = \Phi(T_f)x_0(r) + S^{-1}(T_f)u_d(r,t) + T^{-1}(T_f)u_b(r_b,t). \quad (2.19)$$

The state trajectory is eliminated from the penalized cost index by substituting Equations 2.18 and 2.19 into Equation 2.17, i.e.,

$$J_p[u;x] = J[u;\Phi(t)x_0+S^{-1}(t)u_d+T^{-1}(t)u_b] + P[\Phi(T_f)x_0+S^{-1}(T_f)u_d + T^{-1}(T_f)u_b]. \quad (2.20)$$

Simplification of Equation 2.20 yields the standard quadratic form

$$J_p[u] = J_0 + \langle c,u\rangle + \frac{1}{2}\langle u,Au\rangle, \tag{2.21}$$

where

$$\begin{aligned} J_0 = c_0 &+ \langle(\Phi^*(t)c_2 - \Phi^*(T_f)\Psi_1^*[W^*+W])x_D, x_0\rangle \\ &+ \langle(\frac{\Phi(t)}{2} + \Psi_1\Phi(T_f))x_0,\ (\frac{N\Phi(t)}{2} + W\Psi_1\Phi(T_f))x_0\rangle \\ &+ \langle x_D, Wx_D\rangle, \end{aligned} \tag{2.22}$$

$$c = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \end{bmatrix} = \begin{bmatrix} c_1^d + [S^{-1}(t)]^* c_2 + [\frac{1}{2}[S^{-1}(t)]^*(N^*+N)]\Phi(t)x_0 \\ \quad + [[S^{-1}(T_f)]^*\Psi_1^*[W^*+W]\Psi_1]\Phi(T_f)x_0 \\ \quad - [S^{-1}(T_f)]^*\Psi_1^*[W^*+W]x_D \\ c_1^b + [T^{-1}(t)]^* c_2 + [\frac{1}{2}[T^{-1}(t)]^*(N^*+N)]\Phi(t)x_0 \\ \quad + [[T^{-1}(T_f)]^*\Psi_1^*[W^*+W]\Psi_1]\Phi(T_f)x_0 \\ \quad - [T^{-1}(T_f)]^*\Psi_1^*[W^*+W]x_D \end{bmatrix} \tag{2.23}$$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{2.24}$$

with

$$A_{11}=M_d+[S^{-1}(t)]^*NS^{-1}(t)+2[S^{-1}(T_f)]^*\Psi_1^*W\Psi_1S^{-1}(T_f), \qquad (2.25)$$

$$A_{12}=[S^{-1}(t)]^*NT^{-1}(t)+2[S^{-1}(T_f)]^*\Psi_1^*W\Psi_1T^{-1}(T_f), \qquad (2.26)$$

$$A_{21}=[T^{-1}(t)]^*NS^{-1}(t)+2[T^{-1}(T_f)]^*\Psi_1^*W\Psi_1S^{-1}(T_f), \qquad (2.27)$$

$$A_{22}=M_b+[T^{-1}(t)]^*NT^{-1}(t)+2[T^{-1}(T_f)]^*\Psi_1^*W\Psi_1T^{-1}(T_f) . \qquad (2.28)$$

The necessary condition for u* to be the element of U which minimizes J_p is that the gradient (utilizing the Frećhet derivative) of J_p with respect to the control u vanish at u*. Thus

$$\left.\frac{\partial J_p}{\partial u}\right|_{u^*} = g(u^*) = \left.\begin{bmatrix}\frac{\partial J_p}{\partial u_d}\\ \frac{\partial J_p}{\partial u_b}\end{bmatrix}\right|_{u^*} = c + \frac{1}{2}[A+A^*]u^* = 0. \qquad (2.29)$$

If A is a self-adjoint operator, then Equation 2.29 yields

$$g(u^*) = c + Au^* = 0 . \qquad (2.30)$$

From Equations 2.25 through 2.28 it follows that for A to be self-adjoint, M_d, M_b, N, Ψ_1, and W must be self-adjoint. If A is self-adjoint then the Hessian of J_p given by

$$\frac{\partial^2 J_p}{\partial u^2} = A, \qquad (2.31)$$

is positive definite. Consequently Equations 2.30 and 2.31 are necessary and sufficient conditions for u* to exist and be the unique optimal control for the penalized cost index J_p.

Methods of Solution

This dissertation is not primarily concerned with formulating necessary conditions for optimality, but rather in developing practical methods for solving distributed parameter optimal control problems. Thus, a brief introduction of the basic optimization methods is warranted. In the optimization literature two basic classifications for the methods of solution have evolved. These categories are generally referred to as the direct and the indirect methods.

Indirect methods

Indirect methods are those methods which determine the optimal control by indirectly solving (in most cases iteratively) the operator equation

$$g(u) = 0 \ . \tag{2.32}$$

In general Equation 2.32 is used to eliminate the control from the state and costate systems. Once the control is eliminated, the state and the costate systems form the classical two point boundary value problem (TPBV). The optimal control can be determined once this TPBV problem is solved. Most

indirect methods are characterized by an iterative modification of either the boundary conditions and/or the partial differential equations.

Direct methods

Direct methods are those methods which determine the optimal control by directly operating on the cost index J. Based on information concerning J and possibly the gradient of J the direct methods result in an iterative procedure which, hopefully, converges to the optimal control. These methods require an initial guess to start the iteration, and then correct this initial guess in a certain predetermined manner. The various direct methods differ principally in the means used to determine the control correction. The gradient methods which are certainly the most popular of this class of direct methods will be discussed in more detail in the following chapter.

CHAPTER III. INTRODUCTION TO GRADIENT METHODS

Gradient methods are direct optimization methods which utilize derivative information during the iteration. The most well-known of the classical gradient methods are the steepest descent method and the Newton-Raphson method. The steepest descent method is a first order method (i.e., it uses first derivative information) which is characterized by simple logic, stability with respect to the initial guess, and slow convergence near the solution. In contrast to the steepest descent method is the Newton-Raphson method, a second order method which exhibits rapid convergence near the solution, but poor stability with respect to the initial guess. In recent years a class of second generation gradient methods have been developed which combine the simplicity and stability of the first order methods with the convergence properties of the second order methods. The most popular of this class of gradient methods are the conjugate gradient method and the Davidon method. Although, the motivation for each of these two methods is different, their performance is strikingly similar. In fact, these two methods (theoretically) produce identical iterations on quadratic problems (32).

At the present time only the standard steepest descent method has been adapted to the optimization of distributed parameter systems. In this dissertation the numerical

adaptation of the conjugate gradient method and the Davidon method to distributed parameter systems is presented.

In general, gradient methods are employed to design computer algorithms which are used to obtain approximate solutions to optimization problems. These algorithms usually consist of two iterative processes, which are interrelated. The terms "outer loop iterator" and "inner loop iterator" are introduced to denote these two iterative processes. The reasons for this designation will become apparent when the algorithm is introduced.

Before presenting the general gradient algorithm some nomenclature and definitions have to be intrdocued. Let the control, the gradient, the direction of search, and the control correction parameter at the n^{th} iteration be denoted by u_n, g_n, s_n and γ_n, respectively; where $U_n \varepsilon U$ for all $n \geq 0$, $g_n \varepsilon G$ for all $n \geq 0$, $s_n \varepsilon \hat{S}$ for all $n \geq 0$, and $\gamma_n \varepsilon R^1$ for all $n \geq 0$, and where U, G, and $\hat{S}$ are real separable Hilbert spaces. In the cases to be considered spaces G, $\hat{S}$, and U are identical.

Definition 3.1: The outer loop iterator, specified by the particular gradient method employed, implicitly determines the direction of search s_n and explicitly performs the control iteration; and is given by

$$u_{n+1} = O_L(u_n, \gamma_n;\ u_{n-1}, \ldots, u_{n-m}), \tag{3.1}$$

where O_L: $R^1 \times \prod_{i=0}^{n-m} U \to U$, and m denotes the number of back points used in the iteration.

Remarks: 1. The semicolon in Equation 3.1 separates the point at which new data are used from the point at which old data are reused.

2. The iteration formula defined by Equation 3.1 is referred to as a one point iterator with memory (33).

Definition 3.2: The inner loop iterator determines the control correction parameter γ_n, and is given by

$$\gamma_n^{i+1} = I_L(\gamma_n^i, J[u_n+\gamma_n^i s_n],\ g(u_n+\gamma_n^i s_n)), \tag{3.2}$$

where

$$I_L: R^1 x R^1 x U x U \to R^1.$$

Gradient Method Algorithm

The interrelationship between the inner loop and the outer loop iterators is best illustrated in the gradient method algorithm. This algorithm is as follows:

Outer loop iteration

1. For n=0, guess an initial control function u_0.

2. Calculate the gradient of the cost functional $g(u_n) = g_n$ by:
   a. integrating the state system from t_0 to T_f;
   b. integrating the costate system backwards from T_f to t_0.
3. Calculate the direction of search s_n.
4. Inner loop iteration: calculate the control correction parameter γ_n.
5. Calculate the control correction.
6. Test the convergence criteria; if these tests are not satisfied, increase n and repeat computations beginning with step 2.

The logic flow chart for the above algorithm is presented in Figure 3.1.

The various gradient methods differ principally in the means used to determine the direction of search s_n (step 3), and the control correction parameter γ_n (step 4). The conjugate gradient method and the Davidon method are outer loop iterators with memory, whereas the steepest descent method is an outer loop method without memory, i.e., steepest descent always searches in the negative gradient direction. Thus, the conjugate gradient method and the Davidon method are able to utilize the results of previous iterates to improve the direction of search; and hence converge more rapidly than the methods without memory.

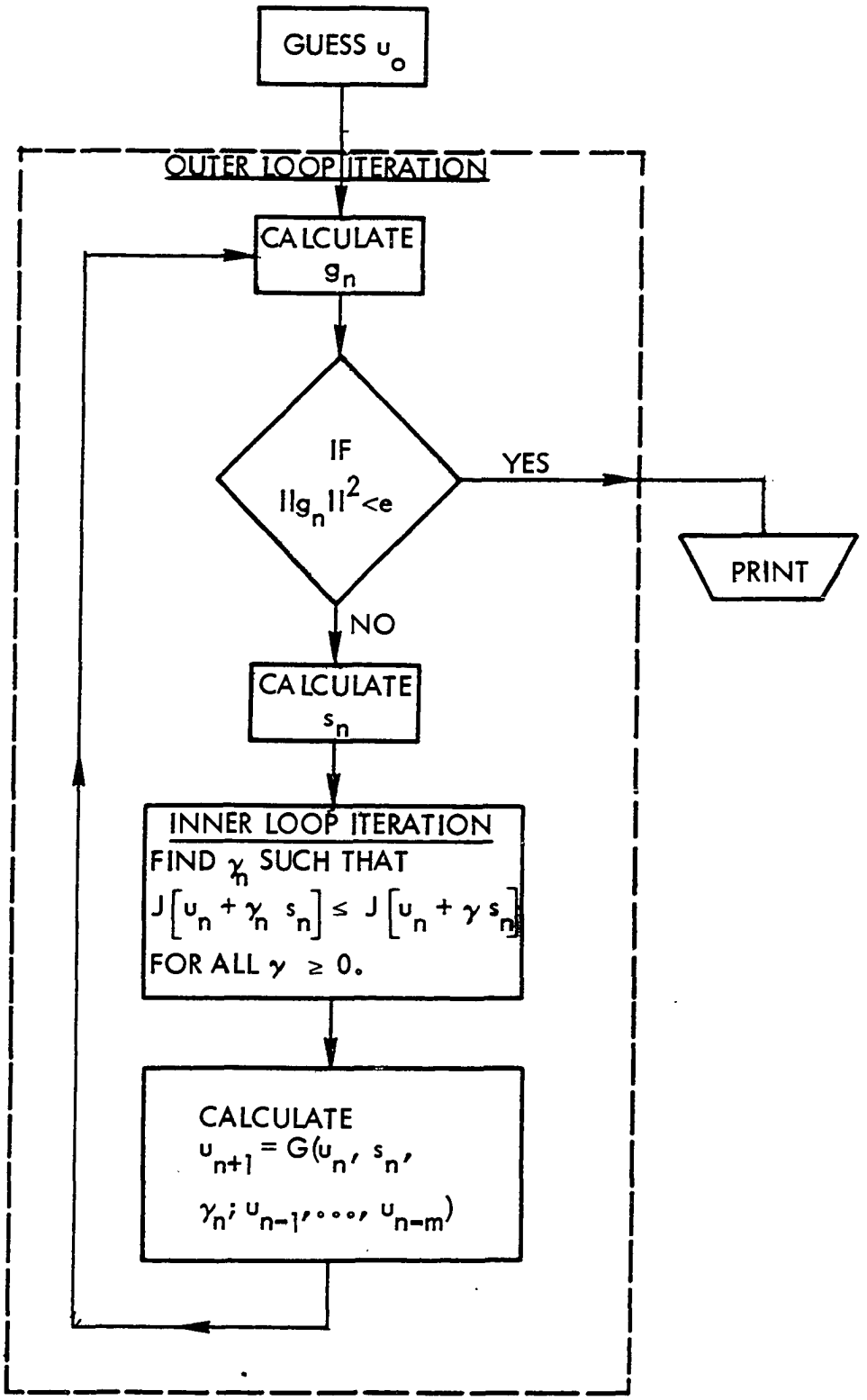


Figure 3.1. The gradient method algorithm

Outer loop iterators

The approximation theory developed in the next chapter applies to gradient methods in general. However, numerical results will be presented for only the following three gradient methods: the steepest descent method, the conjugate gradient method, and the Davidon method. A brief introduction to each of these methods is presented below.

The steepest descent method The steepest descent method is perhaps the oldest of the direct search methods. This method was originally developed for minimizing a function defined on a real Euclidean vector space. An account of this method was given as early as 1847 by Cauchy. Later, it was named the method of gradients by Hadamard. In 1945 the steepest descent method was extended to the case where the function is defined on a Hilbert space (34). More recently Bryson et al. (35, 36) and Kelley (37) have used the steepest descent method to solve lumped parameter optimal control problems. Several authors (9, 28, 29 and 30) have applied the steepest descent method to the distributed parameter optimal control problem.

The basic philosophy of the steepest descent method is very simple. The maximum rate of decrease of J in a neighborhood of an admissible control u_n is in the direction defined by $-g_n$. This direction defines the half-ray $u_{n+1}=u_n-\gamma g_n$, $\gamma \geq 0$. Thus to obtain the maximum decrease in the cost index, the

best local direction of search is in the negative gradient direction; hence, the method is named steepest descent. Consequently the outer loop iterator is given by

$$u_{n+1} = u_n - \gamma_n g_n \ , \tag{3.3}$$

where the control correction parameter γ_n is determined by the inner loop iterator.

It is important to note that in the general case the direction of search s_n defines the direction of maximum decrease in J only for γ_n arbitrarily small. In practice the selection of small control correction parameters leads to excessive iterations. In fact to insure that $\{u_n\}\rightarrow u^*$, γ_n must be bounded away from zero. If $\gamma_N=0$ for some $N\geq 0$, then u_N becomes a fixed point of the outer loop iterator; but g_N is not necessarily the null vector, and hence u_N is not necessarily the minimizing element u^*. The slow convergence of the steepest descent method near the solution can be attributed to the fact that as the iteration converges the gradient tends to the null vector. Hence, the control correction $\gamma_n||g_n||$ becomes excessively small, unless proper precautions are taken in the selection of γ_n. This brief discussion indicates the importance of the inner loop iterator.

The simplicity and the stability of the steepest descent method enables it to be adapted to many difficult, practical

problems. These characteristics are important to practicing engineers, and they often outweigh the slow convergence properties of the steepest descent method.

The conjugate gradient method The conjugate gradient method was originally developed as a method for solving a set of linear algebraic equations; the solution of the set of equations being related to the minimum (maximum) of a certain properly selected cost index (38). In 1954 Hayes (39) extended the original conjugate gradient method to linear operator equations in a Hilbert space. Since then (40) and (41) have also considered the adaptation of this method to the solution of linear operator equations. Fletcher and Reeves (42) then modified the conjugate gradient method and used it to develop a parameter optimization algorithm. Lasdon et al. (43), and Sinnott and Luenberger (44) extended the conjugate gradient method to lumped parameter optimal control problems.

The conjugate gradient method is a gradient method with memory. The motivation for this method is given by the following considerations. Let the set of admissible controls U be a real, separable Hilbert space, i.e., U contains a countable dense subset. The separability insures the existence of at least one linearly independent set of basis vectors $\{s_n\}$, $s_n \varepsilon U$, such that the finite-dimensional subspaces B_n spanned by $\{s_0, s_1, \ldots, s_{n-1}\}$ form an expanding sequence of subspaces,

whose union is the closure of the control space. If for each $n \geq 0$, the inner loop iterator minimizes J over the translation of the one-dimensional subspace defined by $u_{n+1}=u_n+\gamma s_n$, then

$$J[u_{n+1}]=J[u_n+\gamma_n s_n] \leq J[u_n+\gamma s_n] \text{ for all } \gamma \geq 0, \tag{3.4}$$

and

$$J[u_{n+2}]=J[u_{n+1}+\gamma_{n+1} s_{n+1}] \leq J[u_{n+1}+\gamma s_{n+1}] \text{ for all } \gamma \geq 0. \tag{3.5}$$

Thus, two one-dimensional minimizations are sequentially performed over a translation of the subspaces spanned by s_n and s_{n+1}, respectively. The following important question now arises. How can the direction of search s_{n+1} be selected such that the result of this sequence of two one-dimensional minimizations give the same solution as would a two-dimensional minimization over the translation of the two-dimensional subspace spanned by $\{s_n, s_{n+1}\}$. That is, how should s_{n+1} be determined such that

$$J[u_{n+2}] \leq J[u_n+\alpha s_n+\beta s_{n+1}] \text{ for all } \alpha \geq 0 \text{ and } \beta \geq 0. \tag{3.6}$$

The conjugate gradient method generates such an outer loop iterator. This means that the solution obtained by performing a sequence of one-dimensional minimizations over a properly selected set of translated subspaces yields the minimum of the functional over the translated subspace spanned by this set. This method is referred to as the "method of expanding

subspaces".

At the present time there exist two versions of the conjugate gradient method; the original version is developed in (38), and the modified version is developed in (45). Willoughby (46), presents an excellent discussion and comparison of these two versions; and demonstrates numerically that on quadratic functionals these two methods do not produce identical iterations as the theory predicts. Nevertheless, the modified version requires substantially less computation; hence, it will be utilized in what follows.

In the modified conjugate gradient method the direction of search is determined as follows:

$$s_n = -g_n + \beta_n s_{n-1} \ , \tag{3.7}$$

where

$$\beta_n = \frac{\langle g_n, g_n \rangle}{\langle g_{n-1}, g_{n-1} \rangle} \ , \tag{3.8}$$

if n=0, then $\beta_0=0$.

The outer loop iterator for the conjugate gradient method is given by

$$u_{n+1} = u_n + \gamma_n s_n \ . \tag{3.9}$$

The second term on the right hand side of Equation 3.7 is the memory element. This term deflects the direction of search from the negative gradient direction. The modified conjugate gradient method is particularly simple to program,

requires little additional computation and storage in comparison with the steepest descent method, and in general converges much faster than the steepest descent method.

The Davidon method The Davidon method is another popular, second generation gradient method. It was developed by Davidon (47) in 1959, who referred to the method as the "variable metric method". The Davidon method was originally developed as a parameter optimization method. Fletcher and Powell (48) present numerical results, and proofs of convergence and stability for the finite dimensional case. Horwitz and Sarachik (49), and Tokumaru et al. (50), have recently extended Davidon's method to quadratic functionals defined on a real separable Hilbert space; in (50) numerical results are included for a lumped parameter optimal control problem.

The Davidon method like the conjugate gradient method is based on the quadratic approximation. In the quadratic case let A denote the self-adjoint operator generating the quadratic functional, and in the non-linear case let A denote the Hessian operator; then, the Davidon method determines a direction of search

$$s_n = -H_n g_n \ , \tag{3.10}$$

where H_n: U→U, such that the sequence of operators $\{H_n A\}$ converge to the identity operator. Thus, the sequence of

operators $\{H_n\}$ converge to the inverse Hessian A^{-1}. This means that as the Davidon iteration progresses, it becomes similar to Newton's second order method. This fact accounts for the rapid convergence of the Davidon method. The Davidon deflection operator H_n is determined iteratively as follows:

$$H_{n+1}f = H_n f + \langle f,p_n^N\rangle p_n^N - \langle f,q_n^N\rangle q_n^N \ , \tag{3.11}$$

where $f\varepsilon U$,

$$H_0 = I \text{ (or any other idempotent operator)}, \tag{3.12}$$

$$p_n^N = p_n/\sqrt{\langle p_n,y_n\rangle} \ , \tag{3.13}$$

$$q_n^N = q_n/\sqrt{\langle q_n,y_n\rangle} \ , \tag{3.14}$$

$$q_n = H_n y_n, \tag{3.15}$$

$$y_n = (g_{n+1}-g_n)/\gamma_n, \tag{3.16}$$

and γ_n is determined by the inner loop iterative such that

$$J[u_n+\gamma_n s_n] \leq J[u_n+\gamma s_n] \quad \text{for all } \gamma \geq 0. \tag{3.17}$$

The Davidon method generates an outer loop iterator given by

$$u_{n+1} = u_n + \gamma_n s_n, \tag{3.18}$$

where γ_n and s_n are determined from Equations 3.11 through 3.17. The Davidon method contains memory because of the Davidon weighting operator H_n.

As evident from Equation 3.11 the storage requirement of the Davidon algorithm increases with the number of iterations. Thus, even on the large modern digital computers storage problems arise, if a large number of iterations are required to achieve convergence. This drawback of the Davidon method has lead to the practice of restarting the iteration every q iterations. This modification of the Davidon method is referred to as the Davidon[q] method (51). By restarting the Davidon method every q iterations, the storage requirement of the Davidon method is at least bounded. However, when coupled with the inherent storage problems associated with distributed parameter systems, even the Davidon[q] method presents storage problems.

Inner loop iterators

As indicated previously the inner loop iterator determines the amount of control correction. Consequently, the convergence of the inner loop iterator directly influences the convergence of the outer loop iterator. In fact, when the errors due to the various discrete approximations made in solving the problem on a digital computer are considered, it is the inner loop iterator which determines the success or failure of the overall iteration. A detailed discussion of this fact will be deferred until the approximation theory is introduced.

The most popular inner loop iterators are those

which perform a linear minimization in the direction of search s_n. In theory all of these methods converge eventually to the same fixed point. However, in practice this is indeed not the case because of gradient errors. The analysis of the effects of gradient errors on the inner loop iterator will also be given in the next chapter.

The three most popular inner loop iterators are the following:

1. Cubic interpolation based on functional values and directional derivative values (52).
2. Cubic or quadratic interpolation based on functional values (52).
3. Linear interpolation based on directional derivative values, i.e., regula falsi (53).

When there are no errors associated with either the calculation of the cost index J or the gradient g, then method 1 above is cubically convergent, while methods 2 and 3 are quadratically convergent. Thus in this case method 1 is the superior of these three methods. This is not the case when discretization errors are encountered. In fact in this case, method 1 turns out to be the least efficient of these three methods.

General Results for Gradient Methods

The following results are listed for future reference. The cited references contain neither the first nor the only proof available.

Theorem 3.1.: Let U be a real separable Hilbert space with inner product $\langle \cdot,\cdot \rangle$ and norm $||\cdot|| = \sqrt{\langle \cdot,\cdot \rangle}$, let A be a self-adjoint operator defined on U such that

$$m_A||f||^2 < \langle f,Af \rangle < M_A||f||^2,$$

and let J[.] be a quadratic functional defined on U and given by

$$J[u] = J_0 + \langle c,u \rangle + \frac{1}{2}\langle u,Au \rangle$$

with minimum at $u^*=-A^{-1}c$; then, the steepest descent method (54), the conjugate method (original or modified) (41), and the Davidon method (50) with inner loop iterators 1, 2 or 3 generate a sequence $\{u_n\}\to u^*$, and a sequence $\{g(u_n)\}\to 0$.

Theorem 3.2.: (32) For the problem defined in Theorem 3.1 the direction of search vectors s_n of the Davidon method and the conjugate gradient method are positive scalar multiples of each other.

Remark: The proof of Theorem 3.2 presented in (32) is only valid for the finite-dimensional case; however, it can be

generalized with minor extensions.

The above two theorems are particularly significant in this study and will be used repeatedly in what follows. Theorem 3.1 demonstrates that at least theoretically all three of these popular outer loop iterators converge to the minimizing element. Theorem 3.2 presents a connection between the conjugate gradient method and the Davidon method. Due to the generality of these two theorems, they certainly apply to distributed parameter optimal control problems. The proof of convergence of these methods for the general non-linear problem is not a closed question. However, it is at least intuitively clear that if the functional is smooth and convex, then these methods converge to the solution. This argument is founded on the quadratic nature of a smooth convex functional near the minimum. Theorem 3.1 does not ensure, however, that the discretized numerical approximation to the problem defined in Theorem 3.1 will converge. This is significant because it is the discretized version of this problem that is actually solved by the digital computer algorithm. The consideration of the discretized approximation to the optimization problem defined in Theorem 3.1 will be considered in the subsequent chapter.

CHAPTER IV. AN APPROXIMATION THEORY FOR GRADIENT METHODS

Gradient methods are iterative procedures and are therefore only practical when programmed on a high speed digital computer. Thus the original continuous problem is actually replaced by a discrete problem. In the process of transforming the continuous problem into its discrete analog a number of approximations are made which introduce errors. Basically two types of approximations are involved: (1) approximations to elements of a Hilbert space (e.g., the approximation of functions by piecewise polynomials); (2) approximations of operators defined on a Hilbert space (e.g., approximations of differential and integral operators by finite-difference and summation operators, respectively). In addition, there are always errors encountered which are due to numerical round-off.

Until recently the analysis of the effects of these various approximations on the solution of optimization problems has been neglected, in some cases with justification and in others without justification. For example, in early studies of the numerical solutions of parameter optimization problems the effects of round-off were considered important. These effects have been studied from the statistical point of view (55). When finite-difference formulas are employed in parameter optimization problems to calculate the gradient

vector, then the truncation errors of these formulas are encountered. Stewart (56) approaches this problem, in the current fashionable manner, by attempting to eliminate the truncation error. Previous experience (57) by this author reveals that this approach is not an answer, but only a cure and only an approximate cure at best. During the study presented in (57), the need for an analysis of the effects of gradient errors on gradient methods became evident.

Recently two excellent papers (58, 59) have been published which discuss the discretization of the continuous lumped parameter optimal control problem. These papers are concerned with demonstrating the convergence of the solution of the discrete problem to the solution of the continuous problem, as the discretization parameters are refined. From a theoretical point of view this is significant; however, in practice discretization is finite and cannot tend to zero. For as one attempts to let the discretization tend to zero difficulties arise immediately in connection with round-off errors. As a simple example of this phenomenon, consider the approximation of a derivative by a finite-difference formula (e.g., $f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$). If this limiting process is attempted on a finite word length digital computer, the effects of round-off are vivid.

Fortunately, in the case of lumped parameter optimal control problems the effects of truncation error can be controlled. This is largely due to the advanced development of

the state of the art of the numerical solution of ordinary differential equations. It is not meant to imply, however, that the effects of these various approximations can be overlooked in the case of lumped parameter problems. For example a common practice in the numerical solution of lumped parameter optimal control problems is to use a fourth-order Runge-Kutta integration method in the forward integration of the state system, and then to utilize linear interpolation (a first-order method) to obtain the required midpoint values of the state system on the backwards integration of the co-state system. The inconsistency is obvious. The estimates for the errors induced by this type of inconsistent practice on the overall solution is still an open question.

The errors of the discrete approximations involved in the solution of distributed parameter optimization problems on a digital computer are in general larger than in lumped parameter optimal control problems. Hence, the effects of discretization errors upon gradient methods are more pronounced in distributed parameter problems.

The computation of the gradient vector, which for gradient methods is required at least once on every outer loop iteration, primarily consists of the forward integration of the state system and the backwards integration of the co-state system. Thus in the solution of distributed parameter optimal control problems by gradient methods, the repetitive

computation of the gradient vector constitutes a large percentage of the total computing effort. Hence, if high order finite-difference methods are employed in the solution of the state and costate systems, then excessively long computer run times result. If lower order finite-difference methods are used with a small mesh to improve the accuracy, then storage problems arise. In addition for distributed parameter systems, it is a general experience (60) that high order difference formulas are usually quite disappointing in practice. This is in contrast to the situation for lumped parameter systems, where methods like Runge-Kutta achieve remarkable accuracy with little computing effort. The reason for this difference is a basic one: for lumped parameter systems the initial conditions are elements of a real Euclidean vector space, and thus can be represented to a high degree of accuracy on a digital computer, with the error being of the same order as the local round-off error; how accurately the solution at $t+\Delta T$ is computed then depends only upon the utilization of the information available; for distributed parameter systems the initial conditions are elements of a function space (e.g., the Hilbert space L^2), and thus cannot be represented to such a high degree of accuracy on a digital computer, since it would be necessary to store an infinite number of quantities at $t=t_0$; therefore, in computing the solution at $t=t_0+\Delta t$, one is limited by a lack of

needed information. Consequently, only moderately accurate finite-differences methods for the solution of the state and the costate systems are possible with gradient methods. It will be shown that errors introduced by the finite-difference solution of the state and costate systems cause errors in the computation of the gradient vector. Therefore, it becomes necessary to consider the effects of gradient errors on the class of gradient methods.

The Effects of the Discrete Approximations on the Gradient Vector

As indicated in Chapter III the convergence of gradient methods depends strongly on the gradient of the cost index. Therefore, it seems reasonable that the analysis of the properties of the approximate gradient algorithms, such as, convergence, stability, and efficiency, would depend essentially on the analysis of the effects of gradient errors.

In the optimization of distributed parameter systems, all of the approximations (approximation of functions, approximation of operators and round-off) are present, and contribute to gradient errors. In the investigation of these approximations,several results considered below are important.

Let the set of admissible controls U be a real separable Hilbert space, and let $\mathcal{U}$ be a set generated by the application of the discretizing transformation E(h) to the elements of U, that is

$$\mathcal{U} = \{\mu: \mu = E(h)u, \text{ for all } u \in U\}, \tag{4.1}$$

where the discretizing transformation $E(h)$[1] is an evaluation map defined on the nodes N of a net η, and h is the discretization parameter of this net (the definitions of an evaluation map, a net, and the nodes of a net are given in Appendix A).

Example 4.1.: Let $f(t) \in C$ for all $t \in T$, where $T=\{t: a \leq t \leq b\}$, and let the nodes be the set N, where $N=\{t_i: a=t_1, <t_2<\ldots< t_n=b, t_{i+1}=t_i+h\}$. The discretizing transformation is defined, in this case, as

$$E(h)f = \begin{bmatrix} f(t_1) \\ \vdots \\ f(t_n) \end{bmatrix} . \tag{4.2}$$

Let $\tilde{U}$ be a function space generated by the application of an interpolating transformation Q to the elements of $\mathcal{U}$, that is

$$\tilde{U} = \{\tilde{u}: \tilde{u} = Q\mu, \text{ for all } \mu \in \mathcal{U}\} . \tag{4.3}$$

Example 4.2.: Let $\tilde{U}$ be the set of all piecewise quadratic polynomials defined on the set $\mathcal{U}$ determined from Example 4.1.

Some properties of the sets $\mathcal{U}$ and $\tilde{U}$, and the transformations $E(h)$ and Q, which are pertinent to this study, are given in the following lemmas. Proofs of these results are given only in those cases where standard references are not available.

[1]For notational simplicity, the explicit dependence of E on h will be often dropped, i.e., $E \equiv E(h)$.

Lemma 4.1.: The set $\mathcal{U}$ is a finite dimensional linear space.

Proof: Follows from the fact that an evaluation map is a functional on U.

Lemma 4.2.: The set $\tilde{U}$ of piecewise polynomials is a finite-dimensional subspace of U.

Proof: Clearly $\tilde{U} \subset U$, and $\alpha\tilde{u}_1+\tilde{u}_2$ is a piecewise polynomial for all scalars α and vectors $\tilde{u}_1,\tilde{u}_2 \varepsilon \tilde{U}$; hence, $\tilde{U}$ is a subspace of U.

Remarks: (i) $\mathcal{U}$ is complete; hence, with the addition of an inner product it would be a Euclidean space.

(ii) $\mathcal{U}$ and $\tilde{U}$ are isomorphic.

Lemma 4.3.: The interpolating transformation Q is a linear transformation from $\mathcal{U}$ to $\tilde{U}$.

Proof: This lemma follows immediately from the fact that the interpolation formulas defining Q are linear in the function values on the nodes.

Example 4.3.: Consider the following one-dimensional piecewise quadratic interpolation formula

$$Qf(t)=-0.5\ s(1-s)f(I-1)+(1-s)(1+s)f(I)+0.5\ s(1+s)f(I+1)\ ,$$

where $s=(t-t_I)/h$, and f(N) denotes the values of the function on the nodes. Thus

$$Qf(t) = \sum_{j=I-1}^{I+1} a_j f(j),$$

and linearly follows immediately, since

$$Q[\alpha f+\beta g] = \sum_{j=I-1}^{I+1} a_j [\alpha f(j)+\beta g(j)]$$

$$= \alpha\Sigma a_j f(j)+\beta\Sigma a_j g(j)=\alpha Qf+\beta Qg.$$

Thus, even though interpolation between the node points might be quadratic, the operation of interpolation defined on the discrete space $\mathcal{U}$ is a linear transformation.

Lemma 4.4.: For the transformations E(h) and Q,
(i) $E^{-1}(h)$ does not exist, and
(ii) $Q^{-1} = E(h)$.

Proof: (i) obvious
(ii) $E(h)Q\mu=E(h)\tilde{u}=\mu$ because the node points are not altered by Q; hence, $E(h)Q=I$, similarly $QE(h)\tilde{u}=Q\mu=\tilde{u}$; hence $QE(h)=I$.

Remark: On the subspace $\tilde{U}$, E(h) has an inverse, i.e., $E^{-1}(h)=Q$ on $\tilde{U}$.

Lemma 4.5.: The product transformation defined by P=QE(h) is a idempotent operator from U onto the finite-dimensional subspace $\tilde{U}$.

Proof: $P^2u=QEQEu=QEQ\mu=QE\tilde{u}=Q\mu=\tilde{u}$, and

$P^2u=\tilde{u}=Q\mu=QEu=Pu$. Thus $P^2=P$.

In the actual computational process on a digital computer, the discretization is accompanied by the truncation of all but a finite number of digits (approximately fourteen digits in double-precision). This is due to the finite word length of a digital computer. Let T denote the truncation operator, then the Hilbert space U is transformed into the "digital" space D by the transformation TE(h). In addition, when the pseudo binary operations of addition, subtraction, division, and multiplication, which are performed by the digital computer, are considered then this "digital" space is no longer a linear space. For example, because of numerical round-off, the distributive law is no longer exactly satisfied. However, if stable finite-difference methods and double precision arithmetic are utilized, then the effects of round-off become secondary to the other error sources. Thus, for the problems considered in this work $\mathcal{U}$ can be considered to be the digital space. Hence, the discretization process can be thought of as a projection of the continuous problem onto the finite-dimensional subspace $\tilde{U}$. The accuracy of the approximate solution then depends largely on the dimensionality of the space $\mathcal{U}$, and on the interpolation formulas representing Q. The relationships between the spaces U, $\mathcal{U}$, $\tilde{U}$, and D are illustrated in Figure 4.1.

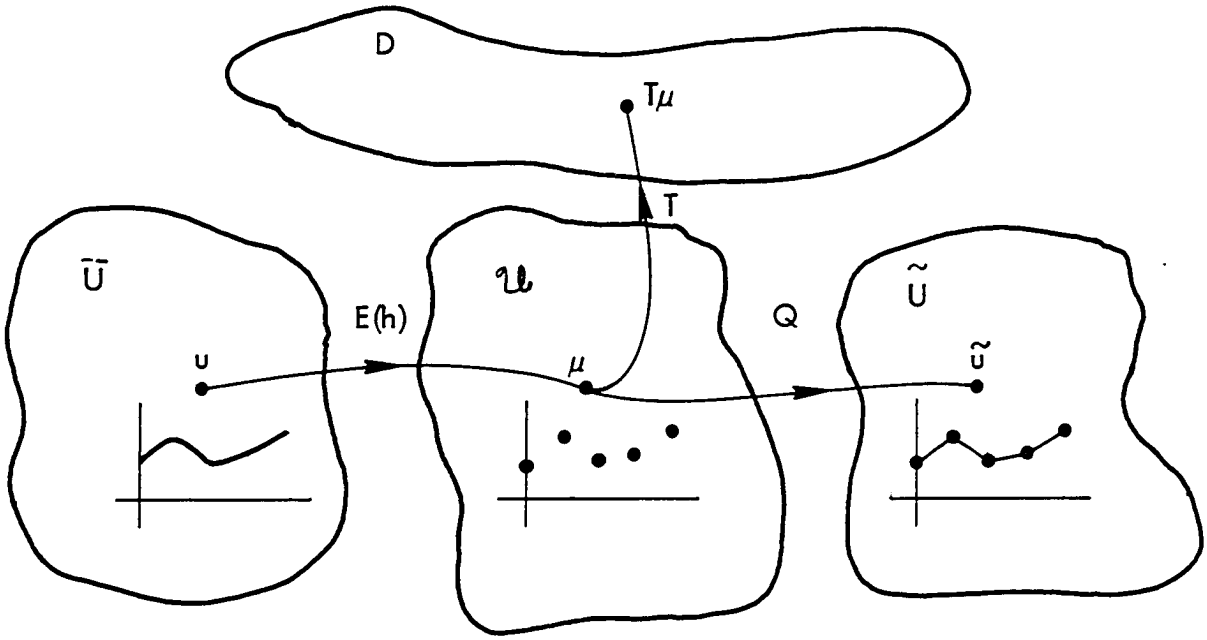


Figure 4.1. The discretization process

Many theoretical results exist for optimization problems in a function space. Unfortunately, the elements of function space and the operators defined on a function space cannot be exactly represented on a digital computer; hence, approximations must be considered. Lemma 4.4 insures that the approximate optimization problem can be considered to be in either the discrete space $\mathcal{U}$ or in the function space $\tilde{U}$. Admittedly, the solution can be calculated at only a finite set of points; however, there can be more information specified about a function than merely its values at a finite set of points, e.g., the functions are polynomial, differentiable,

etc. . Thus, it is felt that the elements of the subspace $\tilde{U}$ give a more complete description of the approximate solution, and solving the problem in this subspace is more in the spirit of the original continuous problem. However, regardless of whether the approximate solution is considered to be in the space $\mathcal{U}$ or in the space $\tilde{U}$, the information which is lost due to discretization cannot be completely regained ($E^{-1}(h)$ does not exist). Therefore, discretization error is caused by the loss of information in the initial and boundary conditions of the state system and in the initial control due to the transformation $E(h)$.

The exact gradient of J for quadratic programming problems is given in Equation 2.30 as

$$g(u) = c+Au . \tag{4.4}$$

Along with this exact gradient the approximate gradient given by

$$\tilde{g}(\tilde{u}) = \tilde{c}+\tilde{A}\tilde{u} . \tag{4.5}$$

is considered. The first question to be answered is the following. How do the approximations of discretization and truncation (round-off is neglected) effect the calculation of the approximate gradient $\tilde{g}(\tilde{u})$?

For the purpose of illustrating how each of these approximations enter into the calculation of $\tilde{g}(\tilde{u})$ consider the following simple problem:

minimize

$$J[u_d(r,t)] = \frac{1}{2}||x(r,T_f)||^2 + \frac{1}{2}||u_d(r,t)||^2, \qquad (4.6)$$

subject to

$$Sx(r,t) = u_d(r,t) \ , \qquad (4.7)$$

$$x(r,0) = x_0(r), \qquad (4.8)$$

$$x_t(r,0) = 0, \qquad (4.9)$$

$$x(0,b) = 0, \qquad (4.10)$$

$$x(1,t) = 0, \qquad (4.11)$$

where

$$S \equiv \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial r^2} \ , \quad ||x||^2 \equiv \int_0^{T_f}\int_0^{R_f} x^2(r,t)\,dr\,dt,$$

and $T_f = 4$, $R_f = 1$. From Equations 2.23, 2.25, and 2.30 the gradient is given by

$$g(u) = u_d(r,t) + [S^{-1}(T_f)]^* [\Phi(T_f)x_0(r) + S^{-1}(T_f)u_d(r,t)], \qquad (4.12)$$

where the term $\Phi(T_f)x_0(r) + S^{-1}(T_f)u_d(r,t)$ represents the forward integration of the state system from t_0 to T_f, and the second term on the right hand side of Equation 4.12 is then given by $[S^{-1}(T_f)]^*[x(r,T_f)]$ which represents the backwards integration of the costate system. This explains the reason for steps (a) and (b) in the gradient algorithm

given in Chapter III. In calculating the gradient on the computer the differential operators S and S* are actually replaced by finite-difference formulas which are truncated approximations of S and S*. This introduces truncation errors. Let $\mathcal{L}$ represent the finite-difference approximations to S, and let $\phi(T_f)Ex_0$ denote the finite-difference solution of the homogenous state system. Then the discrete approximation of Equation 4.12 yields the approximate gradient (discretized)

$$\mathcal{G}(\mu)=E\tilde{g}=Eu_d+[\mathcal{L}^{-1}(T_f)]^*[\phi(T_f)Ex_0+\mathcal{L}^{-1}(T_f)Eu_d] \quad . \qquad (4.13)$$

In Equation 4.13 discretization errors are introduced by the approximation of the initial conditions x_0 by Ex_0 and by the approximation of the control u_d by Eu_d; truncation errors are introduced by the approximation of differential operators by truncated finite-difference operators, which are represented by $\mathcal{L}^{-1}$ and ϕ, respectively. To be consistent the order of the interpolation formulas, represented by Q, should be the same as the order of the finite-difference method, represented by $\mathcal{L}$. Little additional accuracy can be obtained by making the order of Q higher than the order of $\mathcal{L}$, and if the order of Q is lower than the order of $\mathcal{L}$, then interpolation error (see Appendix A) is being needlessly introduced.

The Effects of Gradient Error on Gradient Methods

Let $u_{n+1}=G(u_n,s_n,\gamma_n;\ u_{n-1},\ \ldots,\ u_{n-m})$ represent the exact gradient iteration, then the approximations discussed above yield what will be referred to as the approximate gradient method $\tilde{u}_{n+1}=\tilde{G}(\tilde{u}_n,\tilde{s}_n,\tilde{\gamma}_n;\ \tilde{u}_{n-1},\ldots,\tilde{u}_{n-m})$. The following important questions arise: (1) when there are gradient errors, do the more powerful gradient methods, such as: the conjugate gradient method and the Davidon method, offer advantages or disadvantages over the simpler gradient methods?; (2) given that the convergence of the exact gradient method is assured, under what conditions (if any) will there result convergence of the approximate gradient methods?; (3) if the approximate iteration does converge numerically (in general $\{\tilde{u}_n\}\rightarrow\tilde{u}^*\neq u^*$), at what step should the iteration be terminated in order to insure a reasonable estimate to u^*?; (3) how is this estimate to u^* made and how suboptimal is $\tilde{u}^*$? Before answering these questions some additional nomenclature and definitions have to be introduced. Let $\tilde{u}\in\tilde{U}$ denote the interpolated approximation to $u\in U$, where from Lemma 4.2 $\tilde{U}$ is a finite-dimensional subspace of U. Let $\mathcal{J}$ denote the discrete approximation to J, and let $||\cdot||$ represent the norm of a vector.

Definition 4.1.: (61) If there exists a set $S_{Nc}=\{\tilde{u}: \tilde{u}_{n+1}=\tilde{u}_n, n\geq N\}$, then $\{\tilde{u}_n\}$ is said to be numerically convergent and S_{Nc} is said to be the state of numerical convergence.

Remark: Numerical convergence is different from the standard concept of convergence. This difference is due to the finite word length of a digital computer.

Definition 4.2.: If $J[u^*]=\min_{u\in U} J[u]$, then the optimal control error $||e_u||$ is defined as

$$||e_u||=||\tilde{u}^*-u^*||,$$

where $\tilde{u}^*=\tilde{u}_N \varepsilon S_{Nc}$.

Definition 4.3.: The cost functional error e_J is defined as

$$e_J=|J[u^*]-\mathcal{J}[E(h)\tilde{u}^*]|.$$

Since $J[\tilde{u}^*]$ cannot be computed, the cost functional error is a measure of the suboptimality of the approximate solution.

Gradient errors have two effects on the gradient iteration: (1) direction of search errors in the outer loop iterator, and (2) linear minimization errors in the inner loop iterator. Until more accurate finite-difference methods are developed, it appears that the direction of search errors must be tolerated. However, linear minimization errors can be avoided when the effects of gradient errors on this

phase of the iteration is understood.

If the exact gradient of the cost functional $J[\cdot]$ at $\tilde{u}_n$ is given by $g(\tilde{u}_n)$, then

$$g(\tilde{u}_n) = \tilde{g}(\tilde{u}_n) + e_g(\tilde{u}_n;h), \tag{4.14}$$

where $\tilde{g}(\tilde{u}_n)\varepsilon\tilde{U}$ is the approximate gradient at $\tilde{u}_n$, and $e_g(\tilde{u}_n;h)$ is the gradient error, which as indicated depends on the discretization parameter h of the finite-difference method used in computing the solutions of the state and costate systems.

Many of the following results rely heavily on the linearity of the dynamical system and on the quadratic nature of the cost index. For a well-posed linear dynamical system, there exists a linear transformation, given by Equation 2.18, between the control space U and the state space X. In addition, the discretization of a linear continuous dynamical system results in a linear system of difference equations which when solved yields a linear transformation between the discrete space $\mathcal{U}$ and the discrete state space $\mathcal{X}$. Consequently, the truncation error (on the nodes), which is the difference between these solutions, is also linear in the control. To be more specific, let $g(u)$ denote the discrete approximation of the exact gradient $g(u)$ calculated by the finite-difference solution of the state and costate systems, then

$$\mathcal{g}(u) = \mathcal{e} + \mathcal{A}\mu = \mathcal{c} + \mathcal{A}E(h)u \ . \tag{4.15}$$

The operator $\mathcal{A}$ is linear because the difference equations resulting from the approximation of the linear partial differential equations are linear.

Lemma 4.6.: If the dynamical system is linear and if $J[\cdot]$ is a quadratic functional, then the truncation error in the gradient $e_g(u;h)$ is linear in u. Specifically

$$e_g(u;h) = \xi(h)u + \mathcal{e}_g,$$

where $\xi(h)=E(h)A-\mathcal{A}E(h)$ is a linear operator depending on the discretization parameter h, and $\mathcal{e}_g=E(h)c-\mathcal{e}$.

Proof: The truncation error in the gradient is given by

$$\begin{aligned} e_g(u;h) &= E(h)g(u)-\mathcal{g}(u) \\ &= E(h)[c+Au]-[\mathcal{c}+\mathcal{A}E(h)u] \\ &= E(h)c-\mathcal{e}+[E(h)A-\mathcal{A}E(h)]u \ . \\ &= \mathcal{e}_g + \xi(h)u. \end{aligned}$$

Theorem 4.1.: If the dynamical system is linear, and if J is a quadratic functional, then the approximate gradient $\tilde{g}(\tilde{u})=Q\mathcal{g}(\tilde{u})$ is the exact gradient (apart from round-off) of the quadratic functional

$$\tilde{J}[\tilde{u}]=\tilde{J}_0+\langle\tilde{c},\tilde{u}\rangle+\frac{1}{2}\langle\tilde{u},\tilde{A}\tilde{u}\rangle, \tag{4.16}$$

where

$$\tilde{c} = Q c \text{ , and } \tilde{A} = Q \mathcal{A} E(h)$$

Proof: $\tilde{g}(\tilde{u}) = Q g(\tilde{u})$

$$= Q[c + \mathcal{A} E(h) \tilde{u}]$$

$$= Q c + Q \mathcal{A} E(h) \tilde{u} \text{ ,}$$

which by inspection is the gradient of $\tilde{J}[\tilde{u}]$.

Remark: The inner product $\langle \cdot, \cdot \rangle$ can be calculated exactly (apart from round-off) on the subspace $\tilde{U}$.

Theorem 4.1 is an important result because it implies that even though $\tilde{g}$ is not the gradient of J, or g ; $\tilde{g}$ is the exact gradient of $\tilde{J}$. Therefore, it should be possible to at least minimize $\tilde{J}$. Hopefully, the minimum of this approximate problem will be a satisfactory approximation to the true solution.

The following result will be useful in the analysis of the effects of gradient errors on the inner loop iterators.

Lemma 4.7.: Let γ_n be selected such that $J[u_n+\gamma_n s_n] \leq J[u_n+\gamma s_n]$ for all $\gamma \geq 0$, where s_n is the direction of search. Then γ_n minimizes J along the half-ray $u_n+\gamma s_n$, and is given by

$$\gamma_n = - \frac{\langle g_n, s_n \rangle}{\langle s_n, A s_n \rangle} \quad . \tag{4.17}$$

Proof: Substituting $u_n+\gamma s_n$ into Equation 2.21 yields,

$$J[u_{n+1}]-J[u_0]=\gamma\langle g_n,s_n\rangle + \frac{\gamma^2}{2}\langle s_n,As_n\rangle,$$

The first derivative of the cost change in the direction s_n with respect to γ is

$$\frac{d}{d\gamma}(J[u_{n+1}]-J[u_n]) = \langle g_n,s_n\rangle + \gamma\langle s_n,As_n\rangle.$$

Setting the above equation equal to zero and solving for γ yields

$$\gamma_n = - \frac{\langle g_n,s_n\rangle}{\langle s_n,As_n\rangle} \quad .$$

The second derivative shows that γ_n yields a minimum, i.e.,

$$\frac{d^2}{d\gamma^2}(J[u_{n+1}]-J[u_n]) = \langle s_n,As_n\rangle > m_A||s_n||^2 > 0.$$

By applying Lemma 4.7 it is easily shown that

$$\gamma_n = - \frac{\langle g_n,\tilde{s}_n\rangle}{\langle \tilde{s}_n,A\tilde{s}_n\rangle} \tag{4.18}$$

and

$$\tilde{\gamma}_n = - \frac{\langle \tilde{g}_n,\tilde{s}_n\rangle}{\langle \tilde{s}_n,\tilde{A}\tilde{s}_n\rangle} \quad , \tag{4.19}$$

denote the control correction parameters defined by Equation 3.2 in the direction $\tilde{s}_n$ for J and $\tilde{J}$, respectively. For distributed parameter systems, the operators A and $\tilde{A}$ are, respectively, multiple integral and multiple summation

operators. As a consequence, Equations 4.18 and 4.19 are not generally used in practice. Instead of Equation 4.17, the following methods are usually utilized in the inner loop to determine γ_n:

1. Cubic interpolation based on functional and directional derivative information.
2. Quadratic interpolation based on functional information.
3. Linear gradient interpolation based on directional derivative information (i.e., regula falsi).

In theory these three methods yield the same result. However, method 1 is generally considered to be superior to the other two methods because of its rapid convergence properties. When there exist gradient errors of sufficient magnitude, this is not the case. In fact numerical results indicate that when there are gradient errors then method 1 is the least efficient of these three methods. The convergence of the inner loop iterator is essential to the convergence of the outer loop iterator; hence, the effects of gradient errors on each of these three inner loop methods will be discussed.

Method 1. Cubic interpolation based on functional and directional derivative information

A general description of this method is presented in (52). This method is the most sensitive of these three

methods to gradient error because it requires a close correlation between the gradient and the functional. Reliance on both types of information (gradient and functional values) can cause difficulties if the relative magnitude of the gradient error is large. One reason for the difficulties encountered by this method is that it brackets the minimum in the direction of search (i.e., the iterator determines two scalars γ_n^1 and γ_n^2 such that $\gamma_n^1 \leq \gamma_n \leq \gamma_n^2$) by determining when the directional derivative

$$\frac{dJ}{d\gamma} = \langle g(u_n+\gamma s_n), s_n \rangle \tag{4.20}$$

changes sign, i.e., from negative to positive. Unfortunately due to gradient errors, this method generally does not bracket the minimum. This is illustrated in Figure 4.2.

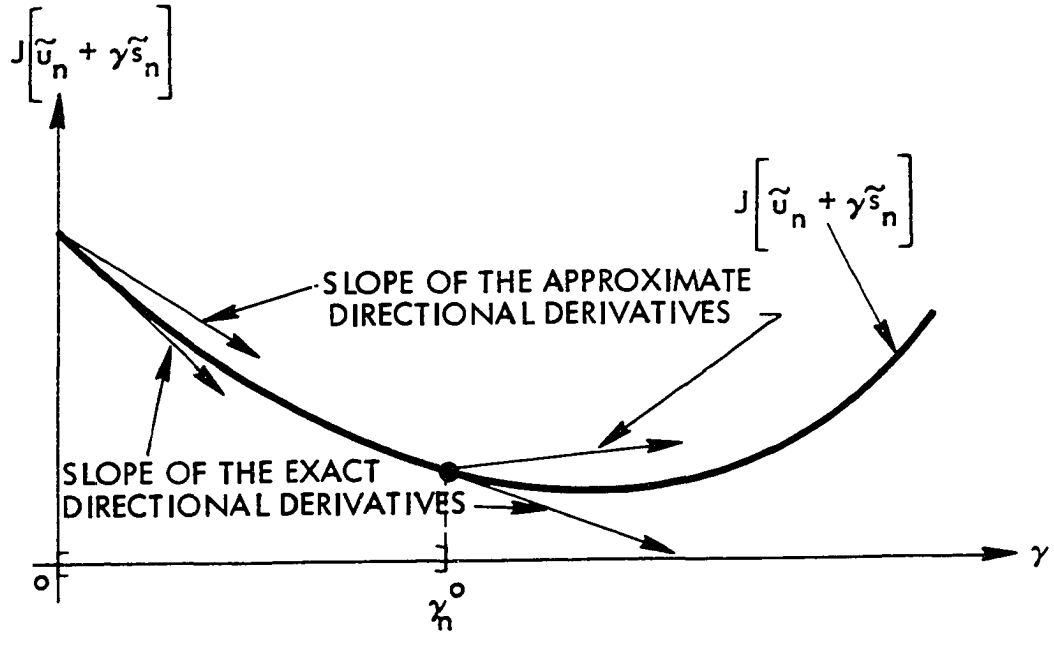


Figure 4.2. Cubic interpolation

As is indicated pictorially in Figure 4.2 the approximate directional derivative at $u_n+\gamma_0 s_n$ can be positive when actually the exact directional derivative is negative. Thus, based on the approximate directional derivative this method would predict that the minimum is in the interval $[0,\gamma_0]$, which is obviously incorrect. This difficulty can be corrected by employing another procedure to bracket the minimum. However, this would only be a minor cure since the interpolation formulas, used by this method, are also based on both types of information. Hence, when there exist considerable gradient errors, this inner loop iterator is not recommended.

Method 2. Quadratic interpolation based on functional values

The fundamental idea underlying this method is the observation that the cost index is nearly quadratic in γ in the direction of search s_n near the minimum. If for fixed u_n and s_n

$$\mathcal{J}[\gamma]=\mathcal{J}[u_n+\gamma s_n]= a_0+a_1\gamma+a_2\gamma^2, \tag{4.21}$$

then by computing $\mathcal{J}[\gamma]$ for $\gamma=\gamma_i$, i=1,2,3, a system of equations is generated from which the coefficients of the assumed polynomial can be obtained. The estimate for γ_n is obtained from the equation

$$\frac{d\mathcal{J}[\gamma]}{d\gamma} = a_1+2a_2\gamma = 0, \tag{4.22}$$

from which

$$\gamma_n \simeq -a_1/2a_2 \,. \tag{4.23}$$

If $\mathcal{J}$ is quadratic then this method determines γ_n in one iteration; however, if $\mathcal{J}$ is not quadratic, then additional logic is required to determine γ_n. The important feature of this method is that it does not depend on the approximate gradient vector. However, since $\tilde{g}$ is not the gradient of $\mathcal{J}$ ($\tilde{g}$ is the gradient of $\tilde{J}$), the directional derivative at the minimum of $\mathcal{J}$ in the direction $\tilde{s}_n$ does not necessarily vanish; hence, the subsequent direction of search is not a conjugate direction and the method of expanding subspaces does not apply. Therefore, if this inner loop iterator is used in conjunction with a conjugate direction method, then rapid convergence cannot be proven. Once again, it is the inconsistency between the gradient and the functional which causes the difficulties. Numerical experience indicates that in the presence of gradient errors this inner loop iterator combined with a conjugate direction method produces slow convergence near the minimum. The slow convergence near the minimum is caused by gradient errors which are more dominant near the minimum (both the exact and the approximate gradients become smaller, in the norm, near the minimum). The main advantage of this inner loop method is that it attempts to minimize $\mathcal{J}$, which may be a better approximation to J than is

$\tilde{J}$. Methods for terminating this iteration will be presented later, since the standard test on the norm of the gradient no longer applies in this case.

Method 3. Linear interpolation of the approximate directional derivative (regula falsi)

Regula falsi has not received widespread application as an inner loop iterator; however, it is probably the oldest of these three methods. This method is similar to Newton's method in that it determines the zero of the gradient rather than the minimum of the functional. When regula falsi is employed as an inner loop iterator, it determines the zero of the approximate directional derivative; hence, the minimum of $\tilde{J}$ in the direction of search. Like method 2 this procedure does not mix the gradient and functional information. Referring to Figure 4.3 the interpolation procedure is as follows: assume

$$\frac{d\tilde{J}}{d\gamma} = a_0+a_1\gamma, \tag{4.24}$$

then

$$\frac{d\tilde{J}}{d\gamma}\Big|_{\gamma=\alpha} = a_0+a_1\alpha, \tag{4.25}$$

and

$$\frac{d\tilde{J}}{d\gamma}\Big|_{\gamma=\beta} = a_0+a_1\beta \ . \tag{4.26}$$

The above relations yield two equations in the unknowns

a_0 and a_1. By solving these equations for a_0 and a_1, an approximate expression for $\frac{d\tilde{J}}{d\gamma}$ is obtained. The control correction parameter $\tilde{\gamma}_n$ is determined from the zero of $\frac{d\tilde{J}}{d\gamma}$, which is given by

$$\tilde{\gamma}_n = (\beta \frac{d\tilde{J}}{d\gamma}\Big|_{\gamma=\alpha} - \alpha \frac{d\tilde{J}}{d\gamma}\Big|_{\gamma=\beta}) / (\frac{d\tilde{J}}{d\gamma}\Big|_{\gamma=\alpha} - \frac{d\tilde{J}}{d\gamma}\Big|_{\gamma=\beta}). \tag{4.27}$$

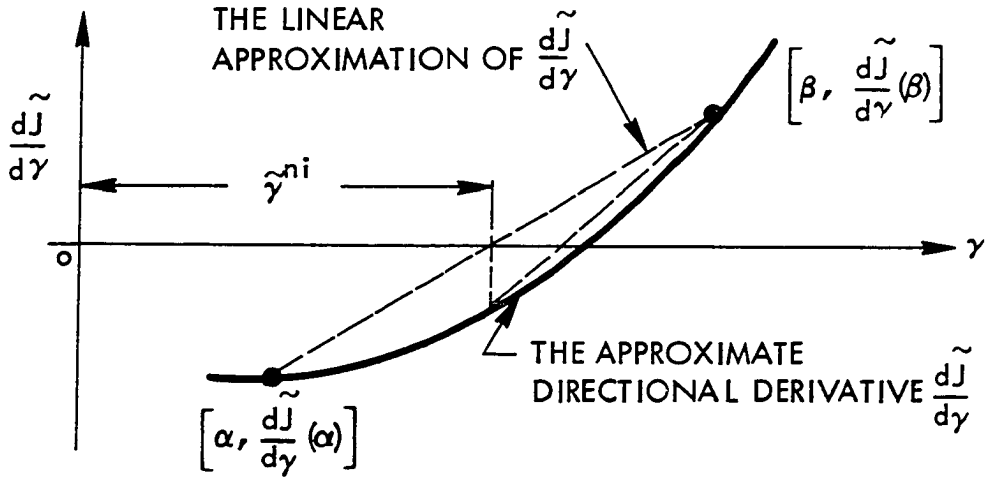


Figure 4.3. The regula falsi iterator

Lemma 4.8.: If $\tilde{J}$ is a quadratic functional, then the regula falsi method determines $\tilde{\gamma}_n$ such that

$$\tilde{J}[\tilde{u}_n+\tilde{\gamma}_n\tilde{s}_n] \leq \tilde{J}[\tilde{u}_n+\gamma\tilde{s}_n] \text{ for all } \gamma\geq 0.$$

Proof: For simplicity let $\alpha=0$, then Equation 4.27 yields

$$\tilde{\gamma}_n = (<\tilde{g}_n,\tilde{s}_n>\beta)/(<\tilde{g}_n,\tilde{s}_n>-<\tilde{g}(\tilde{u}_n+\beta\tilde{s}_n),\tilde{s}_n>)$$

$$= (<\tilde{g}_n,\tilde{s}_n>\beta)/(<\tilde{g}_n,\tilde{s}_n>-<\tilde{c}+\tilde{A}(\tilde{u}_n+\beta\tilde{s}_n),\tilde{s}_n>)$$

$$= (<\tilde{g}_n,\tilde{s}_n>\beta)/(<\tilde{g}_n,\tilde{s}_n>-<\tilde{g}_n+\beta\tilde{A}\tilde{s}_n,\tilde{s}_n>)$$

$$= -<\tilde{g}_n,\tilde{s}_n>/<\tilde{s}_n,\tilde{A}\tilde{s}_n>,$$

and the proof then follows from Lemma 4.7.

Thus, the regula falsi method is a numerical procedure for obtaining the result of Equation 4.19.

Theorem 4.2.: If A is a self-adjoint operator, if the conditions of Theorem 4.1 are satisfied, and if the regula falsi method is used in the inner loop, then the gradient iteration $\tilde{u}_{n+1}=\tilde{G}(\tilde{u}_n,\tilde{s}_n,\tilde{\gamma}_n;\ \tilde{u}_{n-1},\ldots,\tilde{u}_{n-m})$ generates a sequence $\{\tilde{u}_n\}$ which numerically converges to the approximate minimizing element in $\tilde{U}$, specifically

$$\tilde{J}[\tilde{u}^*] = \min_{\tilde{u}\in\tilde{U}} \tilde{J}[\tilde{u}] \ .$$

Remark: The standard convergence proofs for these three gradient methods can be applied to Theorem 4.2 (refer to the references given in Chapter III).

Corollary 4.1.: If the conditions of Theorem 4.2 are satisfied then both the conjugate gradient method and the Davidon method converge in a finite number of iterations to the minimum of $\tilde{J}$.

Proof: The proof follows from the fact that $\tilde{U}$ is a finite-dimensional subspace of U, and from the application of the results contained in (41) and (50).

Theorem 4.2 is significant, since it implies that $\{\tilde{u}_n\}$ generated by $\tilde{G}$ minimizes $\tilde{J}$ and not J nor g. Corollary 4.1 is important, since it insures convergence of the conjugate gradient method and the Davidon method in a finite number of iterations.

Error Estimates

Since $\{\tilde{u}_n\}$ does not minimize J, it is desirable to compute the optimal control error. This, however, is impossible without prior knowledge of the solution u*. Nevertheless, estimates of the optimal control error may be obtained by means of the results given below.

Lemma 4.9.: Let $\nabla_u J$ denote the gradient of a positive definite quadratic functional J defined on a real Hilbert space U, and generated by a self-adjoint operator A, then:

(i) $J_g[u]=||\nabla_u J||^2=||g(u)||^2$ is also a quadratic functional with Hessian 2AA,

(ii) The set defined by $S_g=\{u: ||g(u)||^2=c\}$ is a hyperellipsoid in the Hilbert space U. and,

(iii) If $J_g[u_g^*]=\min_{u\in U} J_g[u]$, then $u_g^*=u^*$.

Proof: (i) $J_g[u]=||g(u)||^2=<g,g>=<c+Au,c+Au>$

$$= <c,c>+2<c,Au>+<Au,Au>.$$

Since $A=A^*$, it follows that $<Au,Au>=<u,A^*Au>=<u,AAu>$. Thus, $J_g[u]=<c,c>+2<c,Au>+\frac{1}{2}<u,2AAu>$, and the Hessian is then 2AA.

(ii) J_g is quadratic from (i).

(iii) $\nabla_u J_g=2Ac+2AAu$; hence, $\nabla_u J_g = 0$ implies that $AAu=-Ac$ and that $u_g^*=-A^{-1}A^{-1}Ac=-A^{-1}c=u^*$.

Lemma 4.10.: Consider the translation defined by $\hat{u}=u-u^*$, where $\hat{J}[\hat{u}]=J[u]$. Then:

(i) $\hat{J}[\hat{u}]=J[u^*]+\frac{1}{2}<\hat{u},A\hat{u}>$.

(ii) $\nabla_{\hat{u}}\hat{J}[\hat{u}]=\nabla_u J[u]$.

Proof: (i) $\hat{J}[\hat{u}]=J[u]=J[\hat{u}+u^*]$

$$= J_0+\langle c,\hat{u}+u^*\rangle + \frac{1}{2}\langle \hat{u}+u^*,A(\hat{u}+u^*)\rangle .$$

Expanding and using the representation of the solution

$$u^* = -A^{-1}c,$$

yields

$$\hat{J}[\hat{u}]=J_0- \frac{1}{2}\langle c,A^{-1}c\rangle + \frac{1}{2}\langle \hat{u},A\hat{u}\rangle .$$

Now it is easily shown that

$$J[u^*]=J_0+\langle c,u^*\rangle + \frac{1}{2}\langle u^*,Au^*\rangle$$

$$=J_0-\langle c,A^{-1}c\rangle + \frac{1}{2}\langle A^{-1}c,AA^{-1}c\rangle$$

$$= J_0 - \frac{1}{2}\langle c,A^{-1}c\rangle ,$$

and thus,

$$\hat{J}[\hat{u}]=J[u^*] + \frac{1}{2}\langle \hat{u},A\hat{u}\rangle .$$

(ii) $\nabla_{\hat{u}}\hat{J}=A\hat{u}=A(u-u^*)=Au-Au^*=Au-A(-A^{-1}c)$

$$=Au+c=\nabla_u J.$$

Theorem 4.3.: The vectors defined by $||\hat{u}_u||^2=\max_{\hat{u}\in S_{\hat{g}}}||\hat{u}||^2$ and $||\hat{u}_1||^2=\min_{\hat{u}\in S_{\hat{g}}}||\hat{u}||$ are eigenvectors of A^2 (i.e., AA) with eigenvalues M_A^2 and m_A^2, respectively.

Proof: By using a Lagrange multiplier λ the proof of this theorem can be formulated as an optimization problem, i.e., extremize $||\hat{u}||^2$ subject to the constraint $S_{\hat{g}}$. This constrained problem can be reformulated as an unconstrained problem by considering the functional

$$f[\hat{u},\lambda]=||\hat{u}||^2 + \lambda(c-||A\hat{u}||^2).$$

Then, the gradient of f is given by

$$\nabla f[\hat{u},\lambda] = \begin{bmatrix} \frac{\partial f}{\partial \hat{u}} \\ \\ \frac{\partial f}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} 2\hat{u}-2\lambda A^2\hat{u} \\ \\ c-||A\hat{u}||^2 \end{bmatrix},$$

where differentiation is in the Frećhet sense.

By setting $\nabla f=0$, one obtains

$$A^2\hat{u} = \frac{1}{\lambda}\,\hat{u} .$$

Therefore, the vectors $\hat{u}_u$ and $\hat{u}_l$ which extremize $||\hat{u}||^2$ subject to the constraint $S_{\hat{g}}$ are eigenvectors of A^2. Since, M_A and m_A are spectral bounds for A, it follows that M_A^2 and m_A^2 are spectral bounds for A^2 (since, $A^2e=AAe=A\lambda e=\lambda Ae$ $=\lambda\lambda e=\lambda^2 e$).

Theorem 4.4.: Let $||g(\tilde{u}_N)||^2$ denote the exact normed gradient squared of $J[\cdot]$ at $\tilde{u}_N$. Then,

$$\frac{||g(\tilde{u}_N)||}{M_A} < ||\tilde{u}^*-u^*|| < \frac{||g(\tilde{u}_N)||}{m_A} .$$

Proof: Let $\hat{u}=\tilde{u}_N-u^*$ and note that since A is self-adjoint, so is AA. By Lemma 4.10, $||g(\tilde{u}_N)||^2=||\hat{g}(\hat{u})||^2=||A\hat{u}||^2=<\hat{u},A^2\hat{u}>$. Thus from Theorem 4.3,

$$m_A^2||\hat{u}||^2 < <\hat{u},AA\hat{u}> < M_A^2||\hat{u}||^2,$$

and

$$m_A^2||\hat{u}||^2 < ||g(\tilde{u}_N)||^2 < M_A^2||\hat{u}||^2,$$

which yields the desired result after taking the square root of

$$\frac{||\tilde{g}(u_N)||^2}{M_A^2} < ||\tilde{u}_N-u^*||^2 < \frac{||g(\tilde{u}_N)||^2}{m_A^2}.$$

From Theorem 4.2, $\{\tilde{u}_n\}$ minimizes $\tilde{J}$; hence, $\{\tilde{g}(\tilde{u}_n)\}\to 0$. This results in a method by means of which $||g(\tilde{u}_N)||$ can be estimated.

Theorem 4.5.: In the state of numerical convergence (see Definition 4.1), the approximate gradient methods resented by $\tilde{u}_{n+1}=\tilde{G}(\tilde{u}_n,\tilde{s}_n,\tilde{\gamma}_n;\ \tilde{u}_{n-1},..,\tilde{u}_{n-m})$ with $\tilde{\gamma}_n=-<\tilde{g}_n,\tilde{s}_n>/<\tilde{s}_n,\tilde{A}\tilde{s}_n>$, insure that $\tilde{g}(\tilde{u}_N)=0$. Thus, $||g(\tilde{u}_N)||=||e_g(\tilde{u}_N;h)||$.

Proof: From Equation 4.14 $g(\tilde{u}_n)=\tilde{g}(\tilde{u}_n)+e_g(\tilde{u}_n;h)$. Now $\{\tilde{u}_n\}$ minimizes $\tilde{J}$ (Theorem 4.2) which implies that $\{\tilde{g}(\tilde{u}_n)\}\to 0$ with n.

Thus $g(\tilde{u}_n)=e_g(\tilde{u}_n;h)$ for all $n\geq N$.

Combining Theorems 4.4 and 4.5 an estimate of the optimal control error is obtained by means of the following Corollary.

Corollary 4.2: Let $e_g(\tilde{u}_N;h)$ be the gradient error at the N^{th} iteration of $\tilde{u}_{n+1}=\tilde{G}[\tilde{u}_n,\tilde{s}_n,\tilde{Y}_n;\ \tilde{u}_{n-1},\ldots,\tilde{u}_{n-m}]$. Then the estimate

$$||e_g(\tilde{u}_N;h)||/M_A < ||\tilde{u}^*-u^*|| < ||e_g(\tilde{u}_N;h)||/m_A,$$

is obtained

Proof: The proof follows immediately from Theorems 4.4 and 4.5.

Unfortunately, only the projection of the gradient error on the subspace of interpolating functions can be obtained on the computer. Let $Pe_g\varepsilon\tilde{U}$ be the projection of $e_g\varepsilon U$ on the subspace $\tilde{U}$, and let $\tilde{U}^{\perp}$ denote the annihilator of $\tilde{U}$. From the Projection Theorem, (refer to Appendix A) the gradient error is given by

$$e_g(\tilde{u}_n;h) = Pe_g + Y, \tag{4.28}$$

where $Y\varepsilon\tilde{U}^{\perp}$. From the triangle inequality it follows that

$$||e_g(\tilde{u}_n;h)|| \leq ||Pe_g|| + ||Y|| , \tag{4.29}$$

and the estimate

$$||\tilde{u}^*-u^*|| \leq (||Pe_g|| + ||Y||)/m_A \tag{4.30}$$

holds.

From the Projection Theorem, $\tilde{U}+\tilde{U}^{\perp}=U$, and thus, as the dimensionality of $\tilde{U}$ increases (refinement of the discretization) the quantity $||Y||\to 0$. Equation 4.30 yields a practical method by means of which the optimal control error can be estimated.

If method 2 is utilized in the inner loop iteration then $\{\tilde{g}(\tilde{u}_n)\}\neq 0$, and thus, another method for estimating $||g(\tilde{u}_N)||$ is required. Let $\hat{\gamma}_n$ be the control correction parameter for $\mathcal{J}[\mu]$ in the direction $\tilde{s}_n$. Since, method 2 minimizes $\mathcal{J}[\mu]$ in this direction, $\hat{\gamma}_n$ is then given by

$$\hat{\gamma}_n = \frac{<QEg_n, QE\tilde{s}_n>}{<QE\tilde{s}_n, QQE\tilde{s}_n>} . \tag{4.31}$$

If the steepest descent method is employed, or if the other gradient methods are restarted each time an up-hill direction of search occurs, then $\hat{\gamma}_n\to 0$. As noted before this inadvertently creates a fixed point for the iteration, without causing the gradient to vanish. In addition, since $\hat{\gamma}_n$ eventually becomes small, slow convergence results. This property has been noted in numerical results (57). Termination of the iteration occurs when

$$<QEg_n, QE\tilde{g}_n> = 0. \tag{4.32}$$

This implies that

$$\langle QE(\tilde{g}_n+e_g),\ QE\tilde{g}_n\rangle = 0, \tag{4.33}$$

and since $QE\tilde{g}_n=\tilde{g}_n$, Equation 4.33 yields

$$\langle\tilde{g}_n,\tilde{g}_n\rangle = -\langle QEe_g\ ,\tilde{g}_n\rangle\ . \tag{4.34}$$

Now consider the relations,

$$\begin{aligned}||QEg_n||^2 &= \langle QEg_n,QEg_n\rangle = \langle QE(\tilde{g}_n+e_g),QE(\tilde{g}_n+e_g)\rangle \\ &= \langle\tilde{g}_n+QEe_g,\tilde{g}_n+QEe_g\rangle \\ &= ||\tilde{g}_n||^2 +2\langle\tilde{g}_n,QEe_g\rangle+||QEe_g||^2. \end{aligned}\tag{4.35}$$

Substitution of Equation 4.34 into Equation 4.35 yields

$$||QEg_n||^2=||QEe_g||^2-||\tilde{g}_n||^2. \tag{4.36}$$

Using the Projection Theorem and the triangle inequality one obtains the estimate

$$||g_n|| \leq ||QEg_n|| + ||Y||,\quad Y\varepsilon\tilde{U}^{\perp}\ . \tag{4.37}$$

Thus, the estimate

$$||\tilde{u}_N-u^*|| \leq \frac{[||QEe_g||^2-||\tilde{g}_n||^2]^{\frac{1}{2}} + ||Y||}{m_A} \tag{4.38}$$

is obtained for the case where method 2 is used as the inner loop iterator.

An estimate of the cost functional error can also be given in terms of the gradient error and the spectral bounds.

Theorem 4.6: Let J be a quadratic functional defined on U generated by a self-adjoint operator A and by an inner product $\langle \cdot,\cdot \rangle$. Let $\tilde{J}$ be the approximation of J defined on the subspace $\tilde{U}$ generated by $\tilde{A}$ and by the inner product $(\cdot,\cdot)$. Then

$$|J[u^*]-\tilde{J}[\tilde{u}^*]| \leq \frac{1}{2}\sum_{i}^{2}|e^i_{s_p}| + \frac{1}{2}||\hat{u}^*||[||e_g||+||e_g||] + |e_{J_0}|$$

$$+ \frac{1}{2}\{||\tilde{c}||+||e_c||+||e_g||\}\frac{||e_g||}{m_A} \quad ,$$

where $J_0 = \tilde{J}_0 + e_{J_0}$, $c=\tilde{c}+e_c$, and $\langle \cdot,\cdot \rangle = (\cdot,\cdot)+e^i_{s_p}$.

Proof: Let

$$J[u^*]=J_0+\langle c,u^* \rangle + \frac{1}{2}\langle u^*,Au^* \rangle$$

and

$$\tilde{J}[\tilde{u}^*]=\tilde{J}_0+(\tilde{c},\tilde{u}^*) + \frac{1}{2}(\tilde{u}^*,\tilde{A}\tilde{u}^*) \ .$$

The relation

$$\frac{1}{2}\langle u^*,Au^* \rangle = \frac{1}{2}\langle g(u^*),u^* \rangle - \frac{1}{2}\langle u^*,c \rangle,$$

implies that

$$J[u^*]=J_0+ \frac{1}{2}\langle c,u^* \rangle + \frac{1}{2}\langle u^*,g(u^*) \rangle \ .$$

$$=J_0 + \frac{1}{2}\langle \tilde{c}+e_c,\tilde{u}^*+(u^*-\tilde{u}^*) \rangle + \frac{1}{2}\langle u^*,g(u^*) \rangle .$$

Expanding the above equation, taking into account the errors due to the approximate inner products, and taking the absolute value, yields the estimate

$$|J[u^*]-\tilde{J}[\tilde{u}^*]| \leq |e_{J_0}| + \frac{1}{2}|\langle \tilde{c}, u^*-\tilde{u}^*\rangle| + \frac{1}{2}|\langle e_c, \tilde{u}^*\rangle|$$

$$+ \frac{1}{2}|\langle e_c, u^*-\tilde{u}^*\rangle| + \frac{1}{2}|e_{s_p}^1| + \frac{1}{2}|e_{s_p}^2|$$

$$+ \frac{1}{2}|\langle u^*-\tilde{u}^*, g(u^*)\rangle| + \frac{1}{2}|\langle \tilde{u}^*, e_g\rangle|.$$

Using Schwarz's inequality and Corollary 4.2, and by grouping terms properly the desired estimate is obtained.

In practice $\mathcal{J}[E\tilde{u}]$ is computed rather than $\tilde{J}[\tilde{u}]$; however,

$$|J[u^*]-\mathcal{J}[E\tilde{u}^*]| \leq |J[u^*]-\tilde{J}[\tilde{u}^*]| + |e_{\tilde{J}}|, \tag{4.39}$$

where

$$e_{\tilde{J}} = \tilde{J}[\tilde{u}]-\mathcal{J}[E\tilde{u}].$$

The error $e_{\tilde{J}}$ can be eliminated (apart from round-off) by the proper selection of the quaradure formulas. For example, if piecewise quadratic interpolation is employed to determine function values between the node points, then the use of Simpson's quadrature formula over each partition insures that $e_{\tilde{J}}=0$.

Determination of the Parameters in the Error Estimates

The estimates of the optimal control error and the cost functional error are based on the gradient errors and on the spectral bounds m_A and M_A. Hence, in order to use these estimates methods for obtaining these quantities are required.

Gradient error

At the present time two methods have been utilized for estimating $||e_g||$: (1) error bounds in terms of higher order difference, and (2) asymptotic extrapolation. Since the first method is problem dependent and also conservative only the second method will be discussed here.

Asymptotic extrapolation is an attempt to actually compute the gradient error. It is based on the fact that if the approximation to the gradient is of order p, then

$$g(\tilde{u}_N) = \tilde{g}(\tilde{u}_N;h) + h^p \bar{e}_g(\tilde{u}_N) + O(\tilde{u}_N;h^{p+1}), \qquad (4.40)$$

where $\bar{e}_g$ is defined as the magnified error function. Solving for the gradient at $\tilde{u}_N$ by using stepsizes of h and qh, respectively, $0<q<1$, two equations in $g(\tilde{u}_N)$ and $\bar{e}_g(\tilde{u}_N)$ result, which when solved yield

$$||e_g(\tilde{u}_N:h)|| \simeq [1/(1-h^p)]||\tilde{g}(\tilde{u}_N;qh)-\tilde{g}(\tilde{u}_N;h)||. \qquad (4.41)$$

In view of Theorem 4.2, $\tilde{g}(\tilde{u}_N,h)=0$, and thus, the estimate

$$||e_g(\tilde{u}_N;h)|| = [1/(1-h^p)]||\tilde{g}(\tilde{u}_N;qh)|| \qquad (4.42)$$

for the norm of the gradient error is obtained.

For the class of problems considered the lower spectral bound m_A can be determined analytically. The general form of the Hessian operator for this class of problems is

$$A = \alpha+T^*T \ . \qquad (4.43)$$

By Lemma A-1, T*T is self-adjoint with a lower spectral bound of zero. In addition from Lemma A-2, $\alpha+T^*T$ is also self-adjoint with $m_A=\alpha$. This analytical result is certainly an advantage in performing the error analysis on quadratic programming problems. However, in many cases it is not possible to determine analytically either the Hessian operator A, or its spectral bounds. For example, in non-linear problems the operator A does not appear. However, if the quadratic approximation is valid near the minimum of a non-quadratic functional (which is at least convex), then Davidon's method presents a numerical procedure by means of which an approximate Hessian and its spectral bounds can be obtained.

Theorem 5.7.: (50) Let A be a Hessian operator defined on a real separable Hilbert space U. Then there exists a subspace $\bar{U}\subset U$ such that

$$H^n\bar{u} = A^{-1}\bar{u} \text{ as } n\to\infty,$$

for all $u\varepsilon\bar{U}$, where $\{H^n\}$ is the Davidon deflection operator defined by Equations 3.11 through 3.16.

Corollary 4.3.: Let $||H^{n+1}\bar{u}-H^n\bar{u}||<\varepsilon$ for all $n\geq N$, and let m_H^N and M_H^N denote the smallest and the largest eigenvalues of H^N, respectively. Then

$$m_A \simeq 1/M_H^N$$

and

$$M_A \simeq 1/m_H^N .$$

Proof: $H^N A \simeq I$ implies that $M_H^N m_A \simeq 1 \simeq m_H^N M_A$.

Unfortunately, $\tilde{H}^n$ rather than H^n is obtained on the computer, where $\tilde{H}^n \tilde{A} \to I$.[1] Thus, the previous error estimates are valid only in those cases where $m_A \leq m_{\tilde{A}} = 1/M_{\tilde{H}}^N$. If $m_{\tilde{A}}$ decreases under a refinement of the mesh, then one might possibly consider using $m_{\tilde{A}}$ in the error analysis. However, this would probably produce a more conservative error estimate.

Geometric Interpretation of the Error Bounds

Due to gradient errors, one should not expect to obtain the true solution of an optimization problem when gradient methods are employed. Thus, estimation of the errors become an important part of the solution. The error estimates presented in this chapter rely heavily on the quadratic properties of the cost index. These estimates are based upon the following geometrical considerations. Assume a gradient method is employed which ensures the vanishing of the approximate gradient. Then the true gradient at the N^{th} iteration becomes equal to the gradient error. Obviously, if the gradient error can be calculated, then it is possible to

[1] Assuming that method 3 is used in the inner loop.

continue the iteration. However, in general it is much easier to estimate the norm of the gradient error than the gradient error itself. If only the norm of the gradient error is known, then it is impossible to continue the iteration because of the lack of a direction in which to proceed. Now assume that $||e_g||$ can be computed. From Lemma 4.9, the set $S_g=\{u: ||g||^2=c\}$ is a hyperellipsoid, which if orientated properly would have its center at u*. However, since only $||g||$ can be estimated, it is not possible to determine this direction. Nevertheless, the true solution u* must be contained in a hypersphere which has a radius equal to the semi-major axis of the constant gradient (at $\tilde{u}_N$) hyperellipsoids. These considerations are illusted in Figure 4.4.

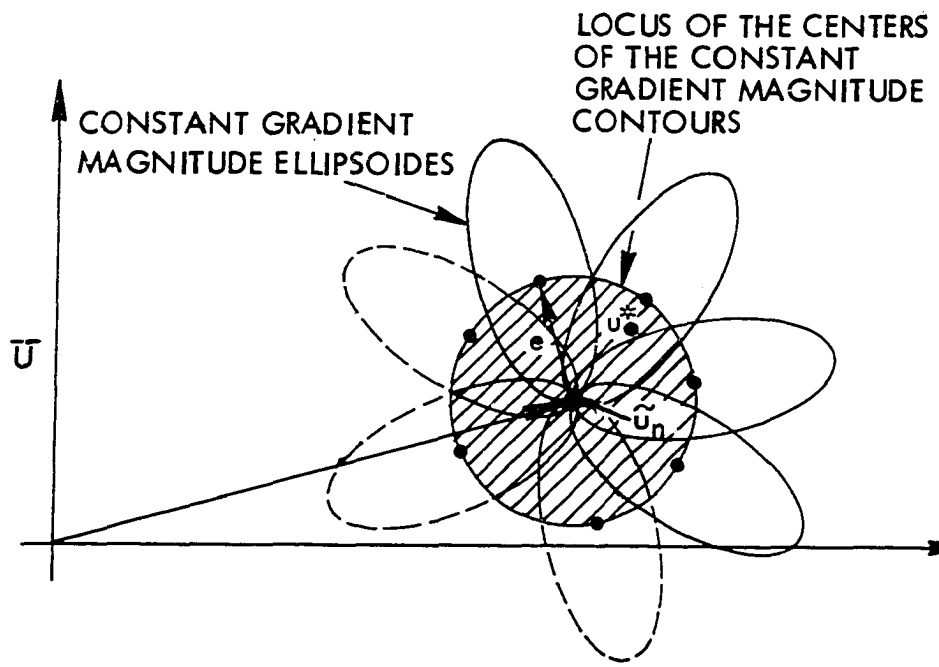


Figure 4.4. Geometrical interpretation of the optimal control error

If the ratio M_A/m_A is large, then these error estimates become conservative. In theory an inner hypersphere can be constructed based on the minor axis of these hyperellipsoids. However, the methods used in estimating the parameters in these error estimates makes these lower bounds questionable. It is worthwhile to note that the constant $\tilde{J}$ contours are in general translated and deformed because of gradient error. This is illustrated in Figure 4.5. The fact that the approximate gradient algorithms only solve the problem in a subspace $\tilde{U}$ of U is illustrated in Figure 4.6.

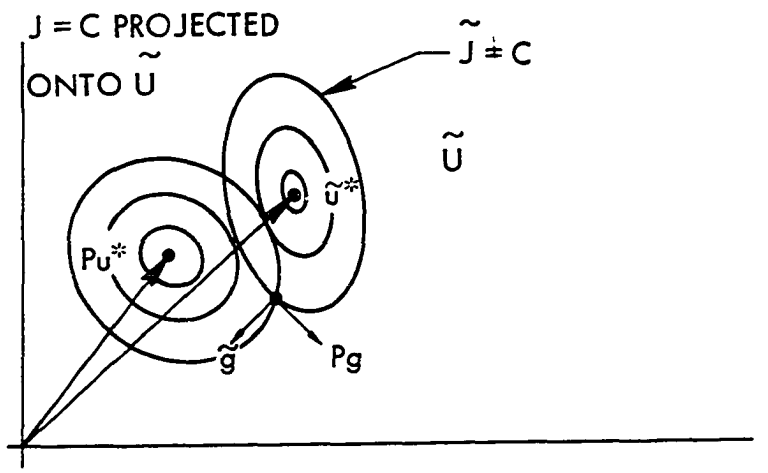


Figure 4.5. Constant cost contours

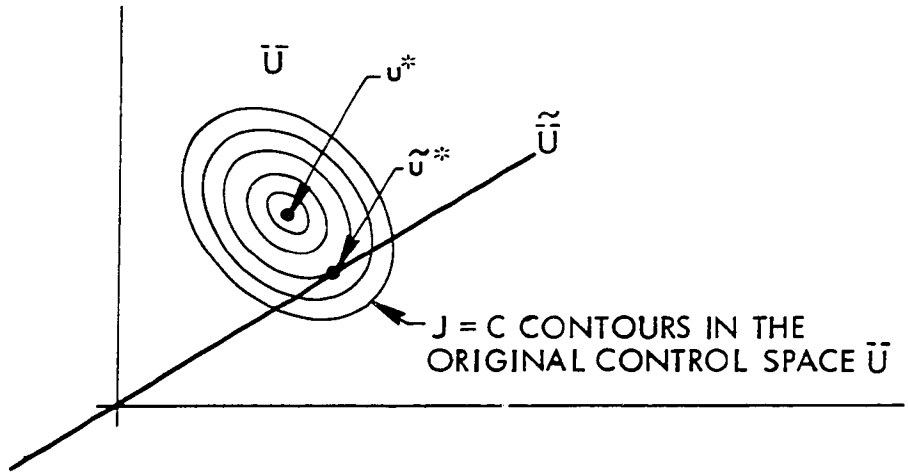


Figure 4.6. Minimization on a subspace

CHAPTER V. NUMERICAL RESULTS

All computations reported in this dissertation were performed on the IBM 360 Model 65 digital computer using the Fortran IV language with double-precision arithmetic. Computation times quoted are the time used by the Central Processing Unit (CPU) during program execution. Although the Central Processing Unit time is the best measure of the computing effort required, it is not precisely reproducible on identical programs due to the multi-programming feature of the system. Storage requirements reported are in terms of array area used in BYTES, which does not include object code storage requirements.

The solution of the state and costate partial differential equations were performed with a standard second order symmetric finite-difference algorithm (62). The multiple quadrature algorithm used in computing the cost functional and the inner products was based on the Gauss-Legendre ten point quadrature formula (53). Piecewise continuous quadratic polynomials were used to obtain function values between the node points. All three types of inner loop iterators described in Chapter IV were employed; however, only the results obtained with method 3 are presented. Numerical results for the modified conjugate gradient method, the Davidon method, and the standard "best step" steepest descent method are presented and compared in Example 5.1. Since the

conjugate gradient method proved superior in terms of CPU time (hence less computer costs) it was utilized on Examples 5.2 and 5.3. In Examples 5.2 the constrained distributed control problem is considered. Example 5.3 presents results for the boundary control problem. The three-dimensional figures presented were generated by the Cal-Comp Digital Incremental Plotter with a subroutine developed by the Iowa State University Computation Center.

Example 5.1.: The unconstrained distributed control of the vibrating string

The unconstrained, fixed time, penalized, minimum energy distributed control of the vibrating string is considered. The problem may be stated as follows:

minimize:

$$J[u_d]=\alpha\int_0^{R_F} x^2(r,T_f)dr + \beta \int_0^{T_f}\int_0^{R_f} u_d^2(r,t)drdt, \quad (5.1)$$

subject to:

$$Sx(r,t)=u_d(r,t), \quad x(r,t) \quad (5.2$$

$$x(r,0)=x_0(r),$$

$$x_t(r,0)=0,$$

$$x(0,t)=0,$$

$$x(1,t)=0,$$

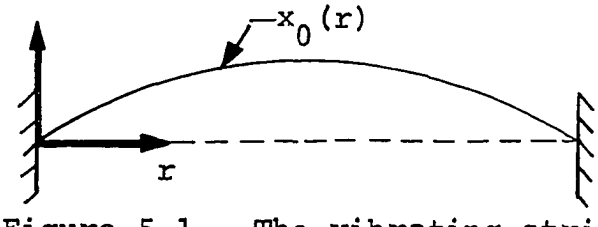


Figure 5.1. The vibrating string

where $S = \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial r^2}$, $x_0(r)=\sin\pi r$, $\alpha=2$, $\beta=0.5$, $R_F=1$, and $T_F=4$. The initial and the boundary conditions are illustrated

in Figure 5.1.

A physical interpretation of the cost functional can be obtained if the inner product for the Hilbert space $L^2[[0,1]x[0,4]]$ is introduced. Let the inner product be given by

$$\langle u,v\rangle = \int_0^{T_f}\int_0^{R_F} u(r,t)v(r,t)\,dr\,dt \ . \tag{5.3}$$

The norm is then

$$||u|| = \sqrt{\langle u,u\rangle} = \left(\int_0^{T_f}\int_0^{R_F} u(r,t)u(r,t)\,dr\,dt\right)^{\frac{1}{2}} \ . \tag{5.4}$$

With this notation the problem may be restated as follows: determine the distributed control $u_d(r,t)\varepsilon L^2[[0,1]x[0,4]]$ which minimizes the sum of the magnitudes of two vectors, (1) the magnitude of the deviation of the string from the equilibrium position at the final time, and (2) the magnitude of the control effort.

For the purposes of illustrating the general theory developed in Chapter II, this problem will be recast in the form given by Equation 2.20.

It can be shown (see Appendix A) that the Green's function for this problem is given by

$$G_F(r,t,\xi,\tau) = \sum_{k=1}^{\infty} \frac{2}{k\pi} \sin k\pi(t-\tau) \sin k\pi\xi \sin k\pi r. \tag{5.5}$$

Thus the formal solution at the final time is

$$x(r,T_f) = \int_0^{R_F} G_F(r,T_f;\xi,t_0)x_0(\xi)d\xi$$

$$+ \int_0^{T_f}\int_0^{R_f} G_F(r,T_f,\xi,\tau)u_d(\xi,\tau)d\xi d\tau, \qquad (5.6)$$

which according to the notation developed in Chapter II becomes

$$x(r,T_f) = \Phi(T_f)x_0 + S^{-1}(T_f)u_d, \qquad (5.7)$$

and as a result the cost index is then

$$J[u_d] = \frac{\alpha}{T_f}||\Phi(T_f)x_0 + S^{-1}(T_f)u_d||^2 + \beta||u_d||^2 . \qquad (5.8)$$

By expanding Equation 5.8 in terms of the inner product, and by employing the definition of the adjoint operator ($\langle Sx,y\rangle=\langle x,S^*y\rangle$) Equation 5.1 becomes

$$J[u_d]=J_0+\langle c,u_d\rangle + \frac{1}{2}\langle u_d,Au_d\rangle, \qquad (5.9)$$

where

$$J_0 = \frac{\alpha}{T_f}||\Phi(T_f)x_0||^2, \qquad (5.10)$$

$$c = \frac{2\alpha}{T_f}[S^{-1}(T_f)]^*\Phi(T_f)x_0, \qquad (5.11)$$

and

$$A = 2\beta + \frac{2\alpha}{T_f}[S^{-1}(T_f)]^*[S^{-1}(T_f)] . \qquad (5.12)$$

Substitution of $\beta=.5$, $\alpha=2$, and $T_f=4$ into Equation 5.12 yields

$$A = 1 + [S^{-1}(T_f)]^* [S^{-1}(T_f)], \tag{5.13}$$

and thus by Lemma A-5, $m_A=1$. The gradient of J is given by

$$\begin{aligned} g(u) &= c + Au \\ &= [S^{-1}(T_f)]^* \Phi(T_f)x_0 + [1+[S^{-1}(T_f)]^* [S^{-1}(T_f)]]u_d \\ &= u_d+[S^{-1}(T_f)]^* [\Phi(T_f)x_0+S^{-1}(T_f)u_d] \\ &= u_d+[S^{-1}(T_f)]^* [x(r,T_f] \ . \\ &= u_d+[\Phi(T_f)]^* [4x(r,T_f)] . \end{aligned} \tag{5.14}$$

Equation 5.6 could be used to compute the cost index and Equation 5.14 could be employed to compute the gradient; however, a brief numerical study of the convergence of the series in Equation 5.5 indicated that a finite-difference method is much more efficient.

A summary of the defining equations and their discrete approximations is given in Table 5.1.

The results of the solution of this problem by the conjugate gradient method (modified), the steepest descent method, and the Davidon method are presented in Table 5.2. These results indicate that for this problem the convergence of the Davidon method is superior to the other two methods. In addition it is apparent from Table 5.2 that both of the second generation methods offer a substantial improvement over the standard "best step" steepest descent method. This can be seen by comparing the approximate gradient magnitudes

Table 5.1. Summary of equations for Example 5.1

Continuous equations	Discrete equations
1. Cost Functional	
$J[u]=\alpha_n\int_0^1 x^2(r,4)dr+\frac{1}{2}\int_0^4\int_0^1 u^2(r,t)drdt$	$[\tilde{u}]=2\sum_{i=1}^{11} a_i\tilde{x}^2(R(i))$
$\alpha_1=2$	$+\frac{1}{2}\sum_{j=1}^{11}\sum_{i=1}^{11} b_{ij}\,\tilde{u}^2(R(i),T(i))$
2. Dynamical System	
$x_{tt} = x_{rr} + u$	$\tilde{x}(i,j)=\tilde{x}(i+1,j-1)+\tilde{x}(i-1,j-1)$ $-\tilde{x}(i,j-2)+h^2\,\tilde{u}(i,j-1)$
$x(r,0) = x_0(r)$,	$\tilde{x}(i,1)=\tilde{x}_0(i)$
$x_t(r,0) = v_0(r)$	$\tilde{x}(i,2)=\frac{1}{2}[\tilde{x}(i-1,1)+\tilde{x}(i+1,1)]$ $+h\tilde{v}_0(i)+\frac{1}{2}h^2\tilde{u}(i,1)$
$x(0,t)=0.0$	$\tilde{x}(1,j)= 0.0$
$x(1,t)=0.0$	$\tilde{x}(11,j)=0.0$
$0\le r\le 1.0$, $0\le t\le 4.0$	$1\le i\le 11$, $1\le j\le 41$, $h=0.1$

Table 5.1 (Continued)

Continuous equations	Discrete equations
3. Adjoint System	
$p_{tt} = p_{rr}$	$\tilde{p}(i,j)=\tilde{p}(i+1,j-1)+\tilde{p}(i-1,j-1)-\tilde{p}(i,j-2)$
$p(r,4)=0.0$	$\tilde{p}(i,1)=0.0$
$p_t(r,4)= 4x(r,4)$	$\tilde{p}(i,2)=\frac{1}{2}[\tilde{p}(i-1,1)+\tilde{p}(i+1,1)]+ 4h\tilde{x}(i,41)$
$p(0,t)=0.0$	$\tilde{p}(1,j)=0$
$p(1,t)=0.0$	$\tilde{p}(11,j)=0$
$t=4-\tau$	
4. Gradient vector	
$g(r,t)=u(r,t)+p(r,t)$	$\tilde{g}(i,j)=\tilde{u}(i,j)+\tilde{p}(i,12-J)$

Note: The (12-J) index is due to backwards integration of the costate system.

Table 5.2. Results for Example 5.1

Iteration number n	Conjugate Gradient Method $\mathcal{J}[\tilde{u}_n]$	Conjugate Gradient Method $\langle \tilde{g}_n, \tilde{g}_n \rangle$	Steepest Descent Method $\mathcal{J}[\tilde{u}_n]$	Steepest Descent Method $\langle \tilde{g}_n, \tilde{g}_n \rangle$	Davidon's Method[q=4] $\mathcal{J}[\tilde{u}_n]$	Davidon's Method[q=4] $\langle \tilde{g}_n, \tilde{g}_n \rangle$
0	0.82497×10^{1}	0.17787×10^{2}	0.82497×10^{1}	0.17787×10^{2}	0.82497×10^{1}	0.17787×10^{2}
1	0.23669×10^{1}	0.63747×10^{1}	0.23669×10^{1}	0.63747×10^{1}	0.23669×10^{1}	0.63747×10^{1}
2	0.81218×10^{0}	0.21307×10^{-3}	0.11382×10^{1}	0.77325×10^{0}	0.81218×10^{0}	0.21307×10^{-3}
3	0.81214×10^{0}	0.25974×10^{-4}	0.88149×10^{0}	0.27711×10^{0}	0.81215×10^{0}	0.69447×10^{-8}
4	0.81215×10^{0}	0.15799×10^{-7}	0.82691×10^{0}	0.33613×10^{-1}	0.81215×10^{0}	0.23487×10^{-12}
5	0.81215×10^{0}	0.24469×10^{-9}	0.81544×10^{0}	0.12046×10^{-1}	0.81215×10^{0}	0.27733×10^{-13}
6	0.81215×10^{0}	0.19422×10^{-11}	0.81292×10^{0}	0.14612×10^{-2}	0.81215×10^{0}	0.22438×10^{-17}
7	0.81215×10^{0}	0.40834×10^{-15}	0.81235×10^{0}	0.52366×10^{-3}	-	-
	CPU time=40.8 sec[a] Storage=22616 BYTES		CPU time=41.5 sec[a] Storage=22616 BYTES		CPU time=97.03 sec Storage=65464 BYTES	

ERROR ANALYSIS: (For the conjugate gradient method only)

Optimal control error: $||\tilde{u}^*-u^*|| \leq 0.11593165\times10^{1}$.

Cost functional error: $|J[u^*]-\mathcal{J}[\tilde{u}^*]| \leq 0.10543401\times10^{2}$.

Where $||e_g(\tilde{u}^*;h)|| \simeq 0.11593165\times10^{1}$, $||\tilde{g}(\tilde{u}^*;qh)||^2 \simeq 0.75600837\times10^{0}$, and $||\tilde{u}^*|| = 0.10489104\times10^{1}$.

CPU time = 50 sec (with error analysis).
Storage = 56936 BYTES (with error analysis).

NOTE: All computations were performed in double precision, only the first five significant figures are reported.

[a]Includes the time required to plot Figure 5.2.

(columns 2, 4, and 6 of Table 5.2) at each iteration. However, the Davidon method required in excess of 150% more CPU time than the modified conjugate gradient method. In addition the Davidon method required 200% more array storage than did the modified conjugate gradient method. Thus, at least for this problem the modified conjugate gradient method appears to be the most efficient of these three methods with respect to computer run-time and storage requirements. In large practical problems the run-time and storage benefits of the modified conjugate gradient method would become an even greater advantage of the method. Since each inner product calculation is essentially a double numerical quadrature, the excessive CPU times of the Davidon[q] method can probably be attributed to the large number of inner products required by the algorithm. However, it might be possible, it extreme care is taken in programming, to make the Davidon[q] method competitive (with respect to storage and CPU time) with the modified conjugate gradient method.

The results presented in columns 1 and 5 of Table 5.2 indicate that the discrete approximation of the cost functional $J[\cdot]$ given by $\mathcal{J}[\cdot]$ does not decrease monotonically, as the conventional optimization theory predicts, but rather increases after the third or fourth iteration. This apparent contradiction is explained by the approximation theory developed in Chapter IV, which showed that the numerical

sequence $\{\tilde{u}_n\}$ generated by the approximate gradient algorithms minimizes $\tilde{J}[\cdot]$ not $\mathcal{J}[\cdot]$, and certainly not $J[\cdot]$. Thus, it is entirely possible, within the context of the approximation theory, for $\{\mathcal{J}[\tilde{u}_n]\}$ not to be monotonically decreasing. The fact that $\{\tilde{u}_n\}$ minimizes $\tilde{J}[\cdot]$ is evident from the decreasing magnitude of the approximate gradient $||\tilde{g}_n||^2$ (columns 2 and 6 of Table 5.2). This brief discussion illustrates the importance of understanding the effects of gradient errors on gradient methods.

The results of the error analysis are also presented in Table 5.2. These results indicate that either the error bounds are conservative or else there are considerable errors introduced by the various approximations involved in the numerical solution. From the results given in Table 5.2 it is observed that the optimal control error is of the same order of magnitude as the norm of the approximate optimal control. In this case it is felt that this does not indicate a conservative error bound, but rather that there is considerable error in the approximate optimal control. This conclusion is based on the observation that after a refinement of the relatively coarse mesh, used in the finite-difference solution of the state and costate systems, the approximated gradient magnitude increased sharply. This indicates substantial gradient errors, in which case large optimal control errors are expected. It also indicates that

for this problem piecewise quadratic functions may not be the best selection for the interpolating functions. Piecewise linear approximating functions were tried but as expected gave even larger estimated control errors. Due to the nature of this particular problem trigonometric approximating function would be the obvious logical choice. A discussion of this consideration will be deferred until the other examples are considered.

The cost functional error estimate is obviously conservative. This of course can be explained by the methods used in deriving this estimate (i.e., the triangle inequality, Schwartz's inequality, etc.).

The initial guessed distributed control, the initial trajectory (1^{st} component), the numerically converged approximate optimal control, and the corresponding optimal trajectory are depicted in Figure 5.2, (a), (b), (c), and (d), respectively.

Example 5.2.: The constrained distributed control of the vibrating string

The constrained, fixed time, fixed terminal state (partial), minimum energy distributed control of the vibrating string is considered. The problem may be stated as follows:

minimize

$$J[u_d]=\beta \int_0^{T_f}\int_0^{R_F} u_d^2(r,t)\,dr\,dt, \qquad (5.15)$$

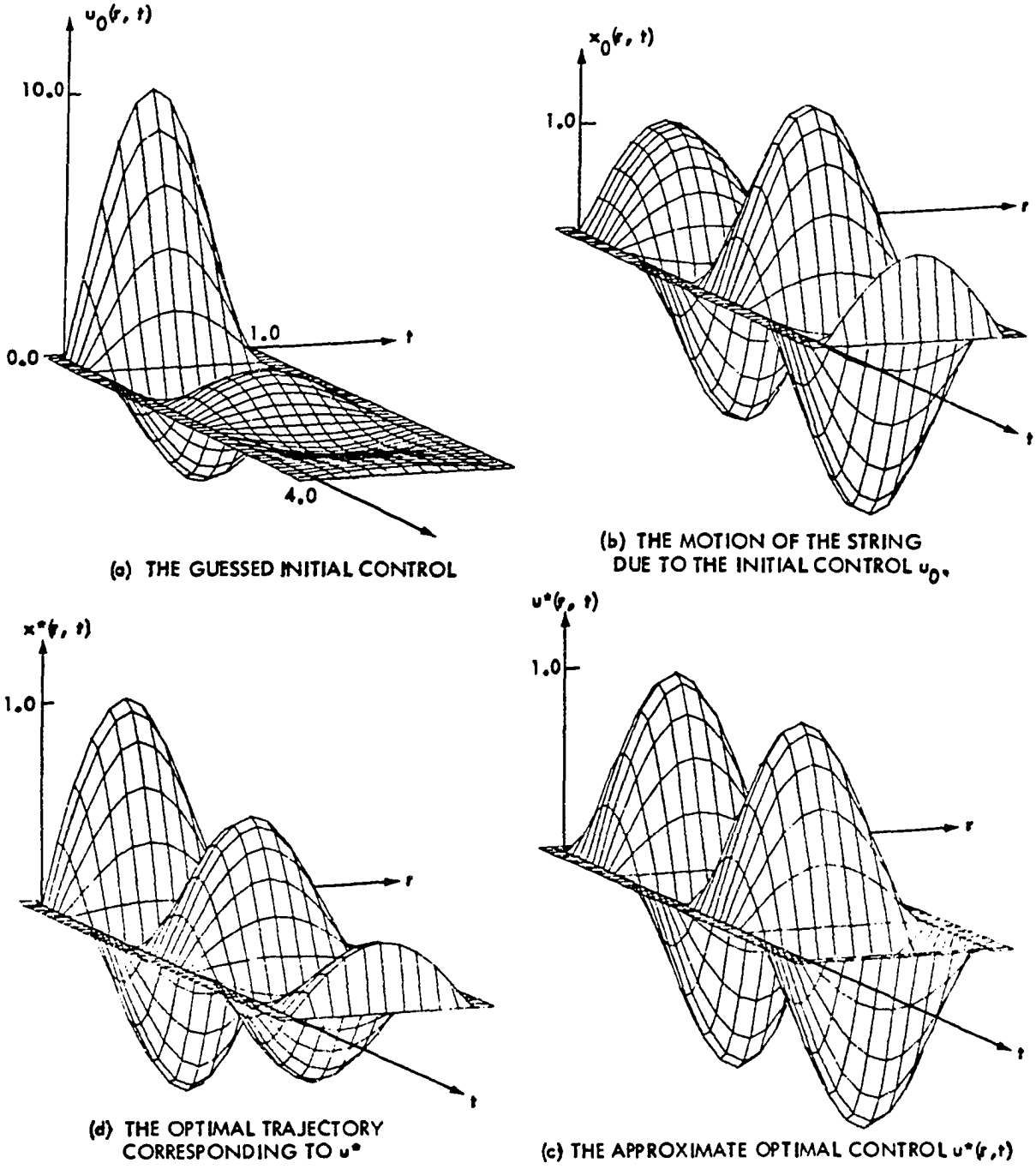


Figure 5.2. The solution to the unconstrained minimum energy distributed control of the vibrating string (R_f = 1.0 and T_f = 1.0)

subject to

$$Sx(r,t)=u_d(r,t), \qquad (5.16)$$
$$x(r,0)=x_0(r),$$
$$x(r,T_f)=0,$$
$$x_t(r,0)=0,$$
$$x(0,t)=0,$$
$$x(1,t)=0,$$

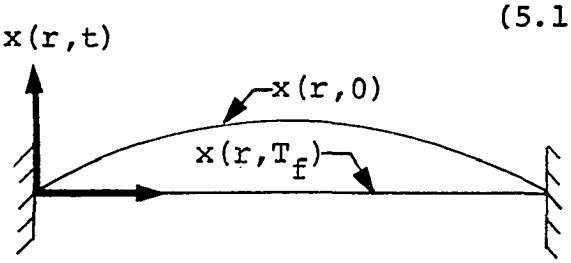


Figure 5.3. The vibrating string

where $S = \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial r^2}$, $x_0(r)=\sin\pi r$, $\alpha=2$, $\beta=0.5$, $R_F=1$, and $T_f=4$. The initial, final, and boundary conditions are illustrated in Figure 5.3. The primary difference between this example and the previous one is that in this case the terminal condition $x(r,T_f)=0$ is included. Since this terminal constraint coupled with the dynamical system constitutes a constraint in the control space U, this problem is not directly solvable by the gradient methods. Thus the penalty function method is employed to alter the form of the problem by replacing the constrained problem by an approximate unconstrained problem. The introduction of the penalty function to account for the terminal constraint yields a new cost functional

$$J_p[u_d]=\alpha_n \int_0^{R_F} x^2(r,T_f)dr + J[u_d], \qquad (5.17)$$

where the penalty constant α_n is arbitrarily chosen. The defining equations and their discrete approximations are then exactly the same as in Example 5.1, and are given in Table 5.1. The Sequential Unconstrained Minimization Procedure is

to solve a sequence of unconstrained problems which converge to the solution of the constrained problem.

Results of the solutions by the modified conjugate gradient method with increasing penalty constants α_n are presented in Table 5.3. The initial guessed control, the initial trajectory, the numerically converged approximate optimal control, and the corresponding optimal trajectory for α_n=100 are depicted in Figure 5.4, (a), (b), (c), and (d). The results of the iteration resulting in Figure 5.4 are given in Table 5.4. From the results presented in Tables 5.3 and 5.4, and from the solution illustrated in Figure 5.4, it appears that the penalty function method offers a practical means for solving constrained problems of this type.

Table 5.3. Penalty constants for the solution of the constrained vibrating string problem

Penalty Constant W	$J=J_p-P[x]$	Constraint Error
2	0.24954254×10^{0}	0.14859240×10^{0}
5	0.55010654×10^{0}	0.52410559×10^{-1}
50	0.10938682×10^{1}	0.10421661×10^{-2}
100	0.11454726×10^{1}	0.27283289×10^{-3}
500	0.11894317×10^{1}	0.11353210×10^{-4}
1000	0.11957950×10^{1}	0.28465427×10^{-5}

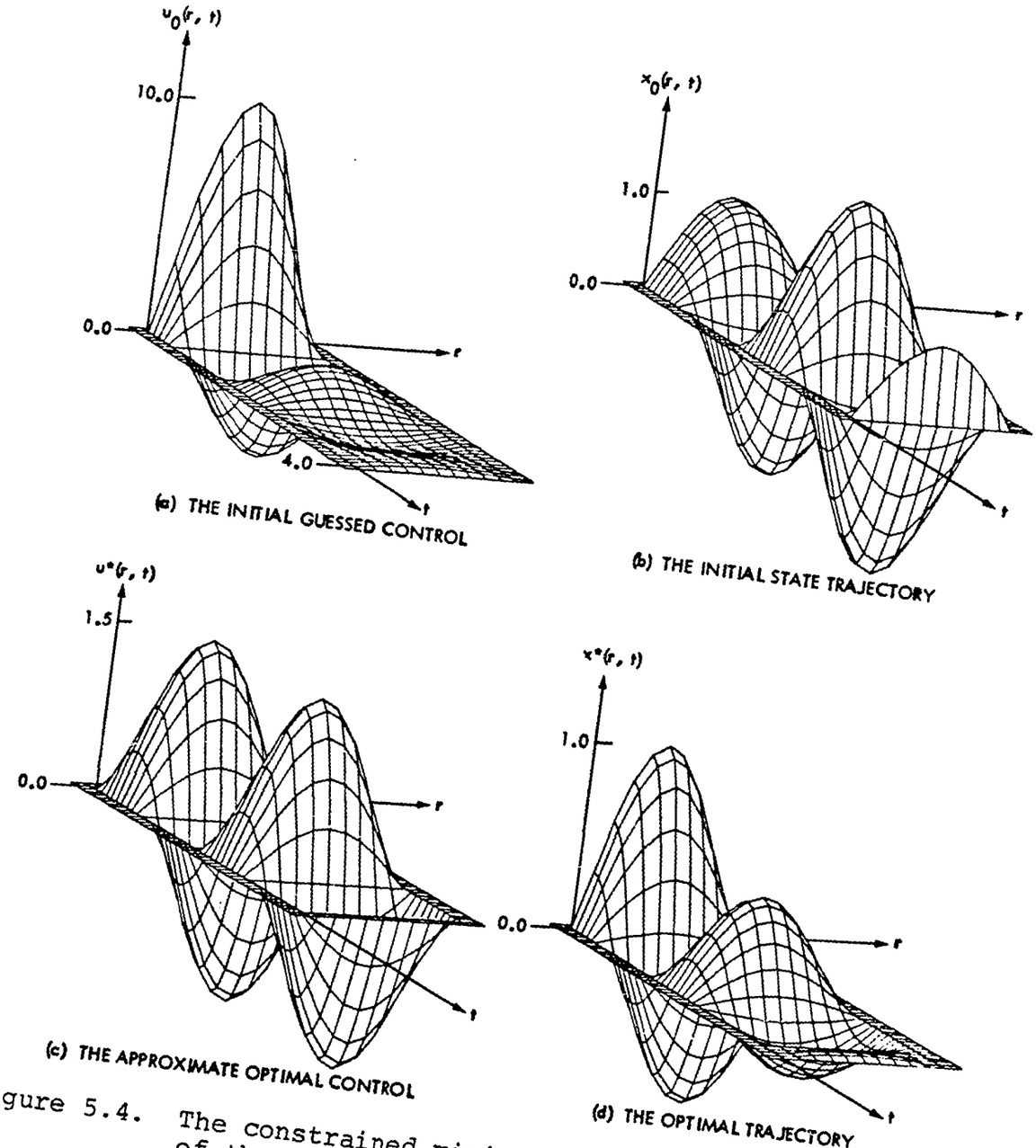


Figure 5.4. The constrained minimum energy distributed control of the vibrating string, $\alpha_n = 100$

Table 5.4. The solution of the constrained vibrating string problem

Iteration number	Modified Conjugate Gradient Method $\mathcal{J}[\tilde{u}_n]$	$\langle \tilde{g}_n, \tilde{g}_n \rangle$
0	0.35919006×10^{2}	0.24209658×10^{4}
1	0.76073334×10^{1}	0.12886314×10^{2}
2	0.11789181×10^{1}	0.52546089×10^{0}
3	0.11727328×10^{1}	$0.22517444 \times 10^{-4}$
4	0.11727398×10^{1}	$0.21432484 \times 10^{-3}$
5	0.11727564×10^{1}	$0.45933457 \times 10^{-6}$
6	0.11727565×10^{1}	$0.55115803 \times 10^{-8}$
7	0.11727560×10^{1}	$0.16968429 \times 10^{-7}$
8	0.11727560×10^{1}	$0.17853686 \times 10^{-11}$
9	0.11727559×10^{1}	$0.44314448 \times 10^{-12}$
10	0.11727559×10^{1}	$0.30146695 \times 10^{-12}$
11	0.11727559×10^{1}	$0.14755488 \times 10^{-15}$

CPU Time=52.6 sec (with plot)

Storage = 140K Total (ARRAY + OBJECT CODE)

$u_0(r,t)=10e^{-t} \sin\pi r \cos\pi t.$

While performing the numerical study of the effects of the penalty constant on the solution, it was discovered that by guessing the initial control to be identically zero (i.e., $u_0(r,t)=0$) the conjugate gradient method converged numerically in exactly one iteration. The results of the iteration for the case where $\alpha_n=5$, $x(r,0)=\sin\pi r$, and $u_0(r,t)=0$ are given

in Table 5.5. Further numerical investigation with different initial conditions and initial guessed controls indicate that the rapid convergence was due to the particular combination of the initial conditions and the initial guessed control (i.e., $x(r,0)=\sin\pi r$ and $u_0(r,t)=0$). Numerical results for the case where $\alpha_n=5$, $x(r,0)=r(1-r)$, and $u_0(r,t)=0$ are given in Table 5.6. It is evident from these results that when the initial condition is polynomial, then numerical convergence from the initial guess $u_0(r,t)=0$ is not obtained in one iteration.

The theoretical implications of these results are interesting. It appears that when the initial conditions are trigonometric (e.g. $x(r,0)=\sin\pi r$) then the solution of the optimization problem is in a finite-dimensional subspace of the control Hilbert space U. Therefore, the infinite dimensional problem is reduced in this special case to a finite-dimensional problem. For example the solution might appear as a finite double Fourier series given by

$$u^*(r,t) = \sum_{n}^{N}\sum_{m}^{M}[a_{nm}\cos nr\cos nt + b_{nm}\cos nr\sin nt + c_{nm}\sin nr\cos nt + d_{nm}\sin nr\sin nt]. \quad (5.18)$$

The parameter optimization problem would then be to determine the Fourier coefficients a_{nm}, b_{nm}, c_{nm}, and d_{nm}. It appears that in this special case the minimizing element of U is contained in the one-dimensional subspace spanned by the

approximate gradient of $\tilde{J}$ at $\tilde{u}_0=0$. In addition the initial guess $\tilde{u}_0$ does not translate the direction of search out of this subspace. Thus only a single one-dimensional minimization is required to obtain the approximate numerical solution. Further comments on how this observation could possibly be used to generate an analytical theory for a special class of problems will be discussed in the next chapter.

Table 5.5. The solution of Example 5.2 with a trigometric initial condition (i.e., $x_0(r)=\sin\pi r$)

Iteration number	Modified Conjugate Gradient Method $\mathcal{J}_p[\tilde{u}_n]$	$\langle\tilde{g}_n,\tilde{g}_n\rangle$
0	0.25092812×10^{1}	0.10533506×10^{2}
1	0.81215934×10^{0}	0.1763511×10^{-28}

Initial Conditions: $x(r,0)=\sin\pi r$, $x_t(r,0)=0$

Initial guessed control: $u_0(r,t)=0$

$\alpha_n=5$, $\beta=.5$

Table 5.6. The solution of Example 5.2 with a non-trigonometric initial condition (i.e., $x_0(r)=r(1-r)$)

Iteration number	Modified Conjugate Gradient Method $\mathcal{J}_p[\tilde{u}_n]$	$\langle\tilde{g}_n,\tilde{g}_n\rangle$
0	0.47825656×10^{-1}	0.18528340×10^{0}
1	0.18137734×10^{-1}	0.23826587×10^{-2}
2	0.18025130×10^{-1}	0.1138757×10^{-5}
3	0.18025621×10^{-1}	0.5359477×10^{-6}
4	0.18025653×10^{-1}	0.35357040×10^{-9}
5	0.18025653×10^{-1}	0.34480603×10^{-13}
6	0.18025714×10^{-1}	0.74813105×10^{-14}

Initial Conditions: $x(r,0)=r(1-r)$, $x_t(r,0)=0$

Initial Guessed Control: $u_0(r,t)=0$,

$\alpha_n=5$, $\beta=.5$

Example 5.3.: The Boundary control of the vibrating string

The fixed time, penalized, minimum energy boundary control of the vibrating string is considered. The problem may be stated as follows:

minimize

$$J[u_b]=\alpha\int_0^{R_F} x^2(r,T_f)dr + \beta\int_0^{T_F} u_b^2(t)dt, \qquad (5.19)$$

subject to

$$Sx(r,t)=0,$$
$$x(r,0)=x_0(r),$$
$$x_t(r,0)=v_0(r),$$
$$Tx(0,t)=u_b(t),$$
$$x(1,t)=0,$$

(5.20)

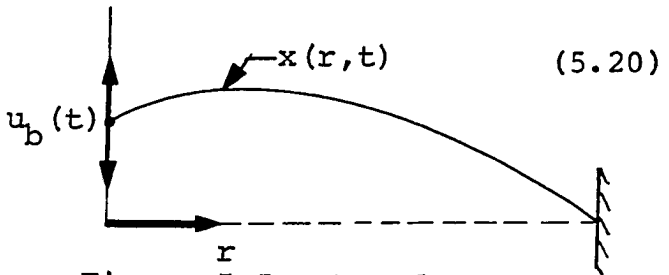


Figure 5.5. Boundary control of the vibrating string

where $S = \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial r^2}$, $T=I$, $x_0(r)=\sin\pi r$, $\alpha=\beta=1$, $R_F=1$, and $T_f=4$. The initial and boundary conditions are illustrated in Figure 5.5. The defining equations and their discrete approximations are given in Table 5.7.

The results of the solution for the minimum effort, boundary control of the vibrating string are given in Tables 5.8 and 5.9, and are illustrated in Figures 5.6 and 5.7. Table 5.8 contains the results of the iteration when the initial guessed control is identically zero (i.e., $u_0(t)=0$). The initial guessed boundary control, the initial trajectory, the numerically converged approximate optimal control, and the corresponding approximate optimal trajectory are shown in Figure 5.6, (a), (b), (c), and (d), respectively. As in Example 5.2, when the initial boundary control was guessed identically zero, the iteration converged in one iteration. The explanation for the rapid convergence is essentially the same as that given in Example 5.2.

Table 5.7. Summary of equations for Example 5.3

Continuous Equations	Discrete Equations
1. Cost Functional	
$J[u]=\int_0^1 x^2(r,T_f)dr+\int_0^4 u^2(t)dt$	$\mathcal{J}[\tilde{u}]=\sum_{i=1}^{11} A_i\tilde{x}^2(R(i),41)+\sum_{i=1}^{11} A_i\tilde{u}^2(T(i))$
2. Dynamical System	
$x_{tt}=x_{rr}$	$\tilde{x}(i,j)=\tilde{x}(i+1,j-1)+\tilde{x}(i-1,j-1)-\tilde{x}(i,j-2)$
$x(r,0)=x_0(r)$	$\tilde{x}(i,1)=\tilde{x}_0(i)$
$x_t(r,0)=v_0(r)$	$\tilde{x}(i,2)=\frac{1}{2}[\tilde{x}_0(i+1)-\tilde{x}_0(i-1)]+hv_0(i)$
$x(0,t)=u(t)$	$\tilde{x}(1,j)=\tilde{u}(j)$
$x(1,t)=0$	$\tilde{x}(11,j)=0.0$
$0\leq r\leq 1.0,\quad 0\leq t\leq 4.0$	$1\leq i\leq 11,\ 1\leq j\leq 41,\ h=0.1$
3. Adjoint System	
$p_{tt}=p_{rr}$	$\tilde{p}(i,j)=\tilde{p}(i+1,j-1)+\tilde{p}(i-1,j-1)-\tilde{p}(i,j-2)$ [a]
$p(r,4)=0.0$	$\tilde{p}(i,1)=0.0$
$p_t(r,4)=2x(r,4)$	$\tilde{p}(i,2)=\frac{1}{2}[\tilde{p}(i-1,1)+\tilde{p}(i+1,1)]+2h\tilde{x}(i,41)$
$p(0,t)=0.0$	$\tilde{p}(1,j)=0.0$
$p(1,t)=0.0$	$p(11,j)=0.0$
$t=4-\tau$	

[a] Integration is performed backwards in time.

Table 5.7 (Continued)

Continuous Equations	Discrete Equations
4. Gradient Vector	
$g(t)=u(t) + \left.\frac{\partial p}{\partial r}\right\|_{r=0}$	$\tilde{g}(j)=2\tilde{u}(j)+\frac{[-p(3,M)+4p(2,M)-3p(1,M)}{2h}$ $M=42-j \qquad h=.1$

Table 5.8. Results for Example 5.3 with $u_0(t)=0$

Iteration number n	Modified Conjugate Gradient Method $\mathcal{J}[\tilde{u}_n]$	$\langle \tilde{g}_n, \tilde{g}_n \rangle$
0	0.50185624×10^{0}	0.88209975×10^{1}
1[a]	0.10027777×10^{0}	$0.31521281 \times 10^{-28}$

ERROR ANALYSIS:

Optimal Control ERROR: $||\tilde{u}^*-u^*|| \leq 0.17718170 \times 10^{0}$

Cost Functional ERROR: $|J[u^*]-\mathcal{J}[\tilde{u}^*]| \leq 0.20936213 \times 10^{0}$,

where $||e_g(\tilde{u}^*,h)|| \simeq 0.35436341 \times 10^{0}$,

$||\tilde{g}(\tilde{u}^*;qh)||^2 = 0.51743680 \times 10^{-1}$

and

$||\tilde{u}^*|| = 0.28581078 \times 10^{0}$.

CPU time = 10.58 sec., Storage = 32400 BYTES.

[a]Convergence in one iteration occurred only when $u_0(t)=0$ was used as the initial control guess.

The results of the iteration for a different initial guess of the control (i.e., $u_0(t)=-10e^{-t}\cos 2\pi t$) are presented in Table 5.9. The initial guessed control, the approximate optimal control, and the corresponding approximate optimal trajectory are illustrated in Figure 5.7, (a), (b), and (c) respectively. The results contained in Table 5.8 and Figure 5.7 indicate that the modified conjugate gradient method will converge (as expected) from a relatively poor initial guess. The converged solution from each of the two different initial guesses are the same, as demonstrated in Tables 5.8 and 5.9 and in Figures 5.6 and 5.7.

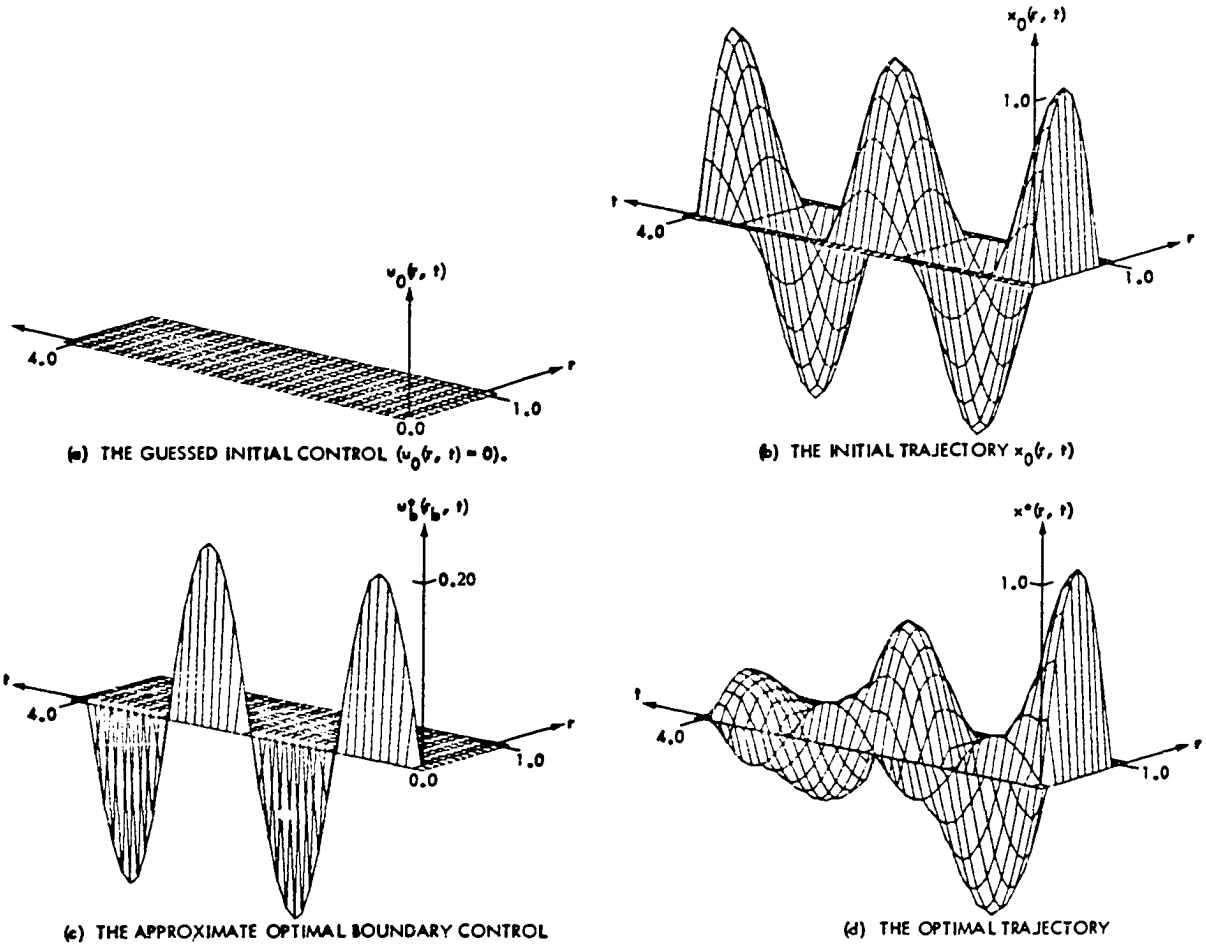
Table 5.9. Results of Example 5.3 with $u_0(t)=-10e^{-t}\cos 2\pi t$

Iteration number	The Modified Conjugate Gradient Method $\mathcal{J}[\tilde{u}_n]$	$\langle\tilde{g}_n,\tilde{g}_n\rangle$	$\langle\tilde{g}_{n+1},\tilde{s}_n\rangle$
0	0.26607469×10^{2}	0.22227374×10^{3}	-
1	0.13385153×10^{2}	0.10305737×10^{3}	0.17763568×10^{-13}
2	0.83627214×10^{1}	0.14073426×10^{3}	$-0.57287508\times10^{-13}$
3	0.24645929×10^{1}	0.38506615×10^{2}	0.82156503×10^{-14}
4	0.45224374×10^{0}	0.91346661×10^{1}	0.38941072×10^{-13}
5	0.21074810×10^{0}	0.20603200×10^{1}	$-0.19428902\times10^{-15}$
6	0.14740533×10^{0}	0.33660356×10^{0}	0.21510571×10^{-15}
7	0.14688323×10^{0}	0.18495487×10^{0}	$-0.13010426\times10^{-16}$
8	0.10460883×10^{0}	0.34394526×10^{0}	0.18561541×10^{-15}
9	0.96151817×10^{-1}	0.61244688×10^{-1}	$-0.49699827\times10^{-15}$
10	0.97951980×10^{-1}	0.19985723×10^{-1}	0.10310762×10^{-15}
11	0.10793993×10^{0}	0.52750740×10^{-1}	0.34640259×10^{-16}
12	0.10831308×10^{0}	0.38984046×10^{-1}	0.22421300×10^{-15}
13	0.10187977×10^{0}	0.43182735×10^{-1}	0.11511516×10^{-15}
14	0.97608394×10^{-1}	0.63656263×10^{-1}	$-0.14007892\times10^{-15}$
15	0.97377092×10^{-1}	0.66172248×10^{-1}	$-0.81185058\times10^{-15}$
16	0.10327465×10^{0}	0.58561917×10^{-2}	0.13216424×10^{-15}
17	0.10309298×10^{0}	0.47066195×10^{-2}	$-0.37100040\times10^{-17}$
18	0.10064508×10^{0}	0.25846288×10^{-2}	0.38916081×10^{-16}
19	0.10031671×10^{0}	0.96233449×10^{-3}	0.17208321×10^{-16}
20	0.10065392×10^{0}	0.37240919×10^{-3}	$-0.66204095\times10^{-16}$
21	0.10036221×10^{0}	0.12510802×10^{-4}	$-0.51698653\times10^{-17}$

CPU Time=34.5 sec (with three dimensional plot).

Storage = 126 K (total).

$u_0(t)=-10e^{-t}\cos(2\pi t)$.



(a) THE GUESSED INITIAL CONTROL ($u_0(t, \tau) = 0$).

(b) THE INITIAL TRAJECTORY $x_0(t, \tau)$

(c) THE APPROXIMATE OPTIMAL BOUNDARY CONTROL

(d) THE OPTIMAL TRAJECTORY

Figure 5.6. The solution of the minimum energy boundary control of the vibrating string

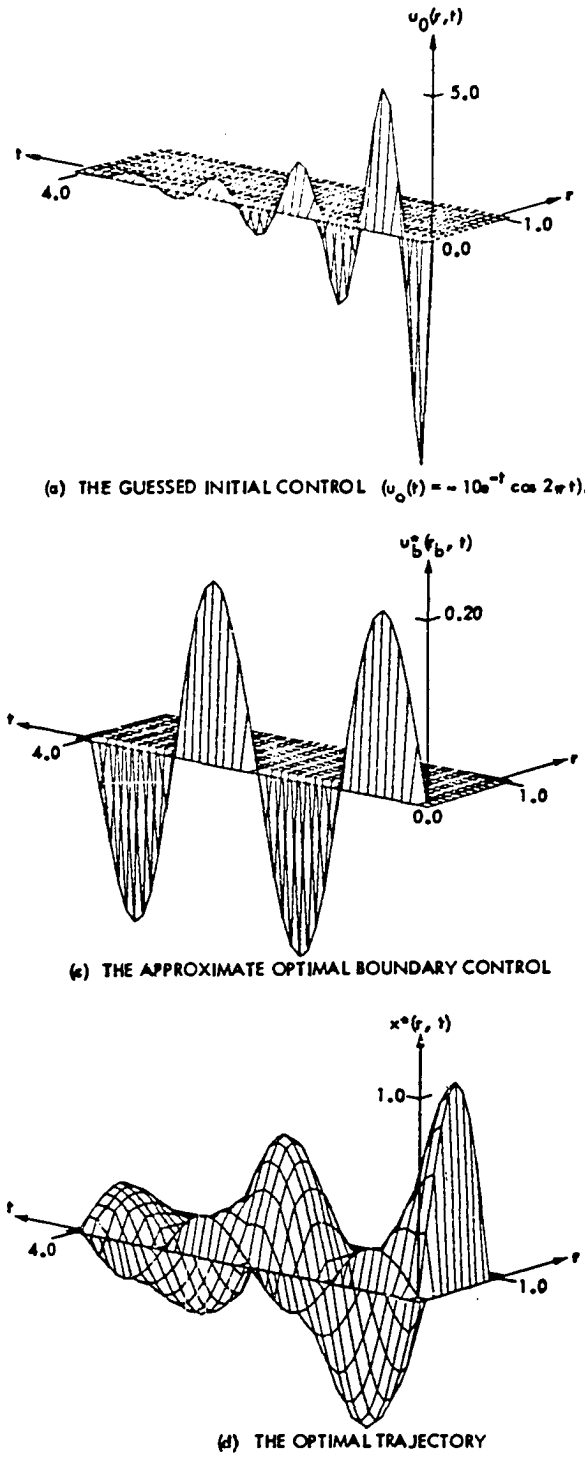


Figure 5.7. The solution of the minimum energy boundary control of the vibrating string

The error analysis presented in Table 5.8 indicates that the optimal control error and the cost functional error are smaller than in Example 5.1. This is probably due to the fact that there is less discretization in boundary control problems than in distributed control problems. As indicated in Tables 5.8 and 5.9 the storage, and the CPU times are also reduced in this boundary control problem. The reduced storage requirements are due to the fact that in boundary control problems the control, the gradient, and the direction of search are all singly subscripted arrays. The reduced CPU times are due to the fact that in boundary control problems there are no double integrals to be approximated.

The accuracy of the inner loop iterator is indicated in column 4 of Table 5.9. The approximate directional derivative at the one-dimensional minimum in the direction $\tilde{s}_n$ is given by $\langle \tilde{g}_{n+1}, \tilde{s}_n \rangle$, and theoretically should be zero. However the numerical accuracy, which is in the order of 10^{-14}, is completely satisfactory. When the other inner loop iterators were utilized $\langle \tilde{g}_{n+1}, \tilde{s}_n \rangle$ was never smaller than 10^{-3}, which demonstrates the validity of the conclusions contained in Chapter IV.

CHAPTER VI. CONCLUDING REMARKS

The original objective of this investigation was to develop practical means of optimizing distributed parameter systems. The gradient methods appeared to be a promising class of methods for solving distributed parameter optimal control problems. However, early numerical results were disappointing. Storage requirements and computation times for test problems indicated that the study of complex distributed parameter systems would probably be beyond the storage capabilities of the present computer system, and certainly beyond a reasonable computer usage budget. Consequently, as is the case in many investigations, the dissertation objectives were modified as work progressed. It soon became apparent that in solving continuous distributed parameter optimization problems on the digital computer the numerous approximations involved in transforming the continuous problem into a discrete problem were causing considerable errors. In order to efficiently obtain an approximate solution, it became evident that the understanding of the effects of these various approximations was essential. Therefore, the modified objectives of developing an approximation theory for the numerical optimization of distributed parameter systems were considered. These objectives have been achieved for a particular class of problems which are of practical significance. Also, even though the original objective was not

achieved, substantial progress was made towards this objective.

An introduction to the optimization of distributed parameter systems was presented in Chapter I, including a discussion of both the theoretical and the numerical results which exist at the present time. Some of the engineering applications for the optimization of distributed parameter systems were also discussed.

A general theory for linear distributed parameter systems was presented in Chapter II. The necessary conditions for a local relative minimum were developed from the functional derivative point of view. It is felt that this development is in the spirit of the subsequent use of gradient methods, since it indicates exactly how the gradient of the cost functional is to be calculated. The penalty function method was introduced to render the constrained problems amenable to gradient methods. A brief discussion of the various methods of solving optimization problems was included.

The three most popular gradient methods were discussed in Chapter III. A comparison between the conjugate gradient method, the Davidon method, and the steepest descent method was given. Three linear minimization methods were introduced, and discussed.

In Chapter IV the analysis of the effects of the many approximations on the solution of the optimal distributed parameter control problem was presented. It was found that in

the numerical optimization of distributed parameter systems by gradient methods the round-off errors are negligible when compared with other approximations involved. Furthermore, it became apparent that the approximations of integral operators (e.g., the cost functional, inner products, etc.) could be made several orders of magnitude more accurate than the approximations of the differential operators. Hence, the two primary error sources were found to be, (1) the approximation of functions, and (2) the approximation of differential operators. It was also shown how these errors effect the computation of the gradient vector, which obviously directly influences the convergence of the approximate gradient methods. In addition, it was demonstrated that the approximate gradient vector is the exact gradient of another quadratic functional, defined on a subspace of the original control space. This result, when coupled with the analysis of the inner loop iteration, leads to necessary and sufficient conditions for the numerical convergence of the approximate gradient methods. It was shown that when method 3 (refer to Theorem 4.2) is utilized in the inner loop then the approximate gradient methods converge to the minimum of $\tilde{J}(\cdot)$, not $J(\cdot)$ nor its discrete analog $\mathcal{J}(\cdot)$. This lead to the concept of optimal control error and of the suboptimality of the approximate solution. Results leading to the estimates of the optimal control error and the suboptimality of the approximate

solution were established. A geometrical interpretation of these estimates was included. It was recognized that these estimates could be used to prove the convergence of the approximate solution to the exact solution as discretization tends to zero. Besides indicating the accuracy of a particular solution, these error estimates yield a discernment into what type of improvements to the approximate solution are feasible.

Numerical results for both the constrained and the unconstrained optimal control of the one-dimensional wave equation were given in Chapter V. Both distributed and boundary control of the wave equation were considered. The standard numerical comparisons between the modified conjugate gradient method, the Davidon method, and the steepest descent method were reported. It is evident from these results that the second generation gradient methods offer a substantial improvement over the steepest descent method, especially with regard to the number of iterations required to achieve numerical convergence. This is particularly significant in the minimization of distributed parameter systems, since each iteration is very costly in terms of computer time. Although the Davidon method converged faster (based on the number of iterations), the modified conjugate gradient method required considerably less storage and computer time to obtain comparable results. Thus, for the class of distributed parameter systems considered the modified conjugate gradient method

appears to be the most efficient method of solution.

It was not expected that the Davidon method would converge more rapidly than the conjugate gradient method, since the theoretical results in (32) indicate that for this class of problems these two iterations should produce the exact same results. In (46) similar differences are reported for the pure and the modified conjugate gradient methods on a quadratic programming problem. The reasons given in (46) for this behavior were based on round-off errors. However, in this case the Davidon method is by far the more complex of the two metnods, and hence should be more sensitive to round-off errors. It is doubtful that in general round-off errors would increase the rate of convergence. Thus the answer to this problem remains an open question.

Due to the discretization processes involved in the solution of the optimization problem on a digital computer the approximate gradient methods do not converged to the exact solution. It was shown that this is due to gradient error. Some authors have argued that when substantial gradient errors are present the more powerful gradient methods should not be used, since the effects of gradient errors might be amplified by these methods. The results contained in this work indicate that this is not necessarily the case. Certainly, it is true that in the presence of gradient errors the gradient methods will not yield the exact solution.

However, if used properly, the more powerful gradient methods, such as the conjugate gradient method or the Davidon method, find the approximate solution much faster; hence, at less cost than the more simple gradient methods.

In conjunction with the theory and the numerical results presented here, there remain many open problems. The extension of both the conjugate gradient method, and the Davidon method to non-quadratic distributed parameter systems is certainly feasible, and the potential results of such extensions appear to be very promising. An extension of the error analysis presented in Chapter IV to nonlinear problems would be of value. Also, consideration of a more conservative cost functional error estimate would be useful. Applications of numerical methods to bounded distributed parameter control problems have not received a great deal of attention to date. The results contained in Tables 5.5 and 5.7 indicate that the application of a Ritz method to the class of problems with trigonometric initial conditions may prove fruitful. In addition, if the initial conditions are not trigonometric, then they could be approximated by an appropriate Fourier series, which could be conveniently truncated to obtain an approximate solution. For problems with linear dynamics an investigation of the penalty constants which circularize the constant cost contours in the control space would be beneficial, since this would increase the rate of convergence

for constrained problems. A more efficient method for programming Davidon's method with emphasis placed on reducing the storage requirements of the method would be a contribution. More sophisticated interpolation methods would most likely reduce the optimal control error and the cost functional error. The utilization of cubic splines present interesting possibilities. Finally, it is suggested that the error analysis and the approximation theory developed in this work be applied to lumped parameter control problems, and to parameter optimization problems (where the gradient vector is obtained from a finite-difference formula).

Although much has already been accomplished, the interest and the activity in this area still remains high; and it is felt that this field still offers numerous possibilities for research.

LITERATURE CITED

1. Butkovskii, A. G. and Lerner, A. Ya. The optimal control of systems with distributed parameters. Automation and Remote Control 21: 472-477. 1960.

2. Butkovskii, A. G. Theory of optimal control for distributed parameter systems. Moscow, Russia, Nayka. 1965.

3. Lions, J. L. Contrôle optimal de systèms governés par des ẽquations aux Dẽrivées Partielles. Paris, France, Dunod Gauthier-Villars. 1968.

4. Koppel, L. B. Introduction to control theory with applications to process control. Englewood Cliffs, New Jersey, Prentice-Hall, Inc. 1968.

5. Denn, M. M. Optimization by variational methods. New York, New York, The McGraw-Hill Book Co. 1969.

6. Brogan, W. L. Optimal control theory applied to systems described by partial differential equations. Ph.D. dissertation. Los Angeles, California, UCLA. 1965.

7. Chaudhuri, A. K. Optimal control computational techniques for distributed parameter systems. Ph.D. dissertation. Gainesville, Florida, Florida University of Gainesville. 1967.

8. Dodd, C. W. An approximate method for optimal control of distributed parameter systems. Ph.D. dissertation. Arizona State University. 1968.

9. Seinfeld, J. H. Optimal control of distributed-parameter systems. Ph.D. dissertation. Princeton, New Jersey, Princeton, University. 1967.

10. Wang, P. K. C. Optimal control of distributed-parameter systems. Transactions of ASME, Journal of Basic Engineering. Series D, 86: 67-79. 1964.

11. Wang, P, K. C. Theory of stability and control for distributed parameter systems (a bibliography). International Journal of Control 7: 101-116. 1968.

12. Butkovskii, A. G., Egorov, A. I., and Lurie, H. A. Optimal control of distributed systems (a survey of Soviet publications). Society of Industrial Applied Mathematics Journal on Control 6: 437-476. 1968.

13. Robinson, A. C. A survey of optimal control of distributed parameter systems. Aerospace Research Laboratories report ARL 69-0177, prepared by the Battelle Columbus Laboratories, Battelle Memorial Institute, Columbus, Ohio. 1969.

14. Butkovskii, A. G. The method of moments in the theory of optimal control of systems with distributed parameters. Automation and Remote Control 24: 1106-1113. 1964.

15. Butkovskii, A. G. Optimum processes in systems with distributed parameters. Automation and Remote Control 22: 13-21. 1961.

16. Butkovskii, A. G. The maximum principle for optimum systems with distributed parameters. Automation and Remote Control 22: 1156-1169. 1962.

17. Egorov, A. I. On optimum control of processes in distributed objects. Applied Mathematics and Mechanics (PMM) 27: 1045-1058. 1963.

18. Lurie, K. A. The Mayer-Bolza problem for multiple integrals and the optimization of the performance of systems with distributed parameters. Applied Mathematics and Mechanics (PMM) 27: 1284-1299. 1963.

19. Balakrishnan, A. V. An operator theoretic formulation of a class of control problems and a steepest descent method of solution. Society of Industrial Applied Mathematics Journal on Control 1: 109-127. 1963.

20. Balakrishnan, A. V. Optimal control problems in Banach-spaces. Society of Industrial Applied Mathematics Journal on Control Series A, 3: 152-179. 1965.

21. Russell, D. L. Optimal regulation of linear symmetrical hyperbolic systems with finite dimensional controls. Society of Industrial Applied Mathematics Journal on Control 4: 276-294. 1966.

22. Axelband, E. I. Optimal control of linear distributed parameter systems. In Leondes, C. T., ed. Advances in control systems. Vol. 7. Pp. 257-310. New York, New York, Academic Press. 1969.

23. Weigand, W. A. and D'Souza, A. F. Optimal control of linear distributed parameter systems with constrained inputs. Transactions of ASME, Journal of Basic Engineering Series D 91: 161-167. 1969.

24. Sakawa, Y. Optimal control of a certain type of linear distributed-parameter systems. IEEE Transactions for Automatic Control AC-11: 35-41. 1966.

25. Magne, F. and Kristansen, T. Optimization of a non-linear distributed parameter system using periodic boundary control. International Journal of Control 10: 601-624. 1969.

26. Lukes, D. L. and Russell, D. L. The quadratic criterion for distributed systems. Society of Industrial Applied Mathematics Journal on Control 7: 101-102. 1969.

27. Phillipson, G. A. and Mitter, S. K. Numerical solution of a distributed identification problem via a direct method. In Balakrishnan, A. V. and Neustadt, L. W., editors. Computing methods in optimization problems. New York, New York, Academic Press. 1969.

28. Sage, A. P. and Chaudhuri, S. P. Gradient and quasi-linearization computational techniques for distributed parameter systems. International Journal of Control 6: 81-98. 1967.

29. Kim, M. M. Successive approximation method in optimum distributed-parameter systems. Journal of Optimization Theory and Applications 4: 40-43. 1969.

30. Denn, M. M. Optimal boundary control for a nonlinear distributed system. International Journal of Control 4: 167-178. 1966.

31. Luenberger, D. G. Optimization by vector space methods. New York, New York, John Wiley and Sons, Inc. 1969.

32. Myers, G. E. Properties of the conjugate gradient and Davidon methods. Journal of Optimization Theory and Applications 2: 209-219. 1968.

33. Traub, J. F. Iterative methods for the solution of equations. Englewood Cliffs, New Jersey, Prentice-Hall, Inc. 1964.

34. Kantorovich, L. V. On an effective method of solution of extremal problems for a quadratic functional. Diklady Akademii Nauk SSSR 48: 483-487. 1945.

35. Bryson, A. E., Jr. and Denham, W. F. Optimal programming problems with inequality constraints. II. Solutions by Steepest Descent. American Institute of Aeronautics and Astronautics Journal 2: 25-34. 1964.

36. Bryson, A. E., Jr., Denham, W. F., and Dreyfus, S. E. Optimal programming problems with inequality constraints. I. Necessary Conditions for Extremal Solutions. American Institute of Aeronautics and Astronautics Journal 1: 2544-2550. 1963.

37. Kelley, H. J. Gradient theory of optimal flight paths. American Rocket Society Journal 30: 947-954. 1960.

38. Hestenes, M. R. and Stiefel, E. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49: 409-436. 1952.

39. Hayes, R. M. Iterative methods of solving linear problems in a Hilbert space. National Bureau of Standards Applied Mathematics Series 39: 71-104. 1954.

40. Daniel, J. W. The conjugate gradient method for linear and nonlinear operator equations. Society of Industrial Applied Mathematics Journal on Numerical Analysis 4: 10-26. 1967.

41. Antosiewicz, H. A. and Rheinbolt, W. C. Conjugate-direction methods and the method of steepest descent In Todd, John, ed. Numerical analysis and functional analysis. Pp. 501-517. New York, New York, McGraw-Hill Book Co., Inc. 1962.

42. Fletcher, R. and Reeves, C. M. Function minimization by conjugate gradients. The Computer Journal 7: 149-154. 1964.

43. Lasden, L. S., Mitter, S., and Waren, A. D. The method of conjugate gradients for optimal control problems. Institute of Electrical and Electronics Engineers Transaction on Automatic Control AC-12: 132-138. 1967.

44. Sinnott, J. F., Jr. and Luenberger, D. G. Solution of optimal control problems by the method of conjugate gradients. Joint Automatic Control Conference Proceedings 1967: 566-574. 1967.

45. Daniel, J. W. Convergence of the conjugate gradient method with computationally convenient modifications. Numerische Mathematik 10:125-131. 1967.

46. Willoughby, J. K. Adaptations of the conjugate gradient method to optimal control problems with terminal state constraints. Ph.D. dissertation. Ames, Iowa, Iowa State University. 1969.

47. Davidon, W. C. Variable metric method for minimization. Argonne National Laboratory Report ANL-5990 (Rev. 2) (Argonne National Lab., Argonne, Ill.) Revised February 1966.

48. Fletcher, R. and Powell, M. A rapidly convergent descent method for minimization. The Computer Journal 6: 163-168. 1963.

49. Horwitz, L. B. and Sarachik, P. E. Davidon's method in Hilbert space. Society of Industrial Applied Mathematics Journal on Applied Mathematics 16: 676-695. 1968.

50. Tokumaru, H., Adachi, N., and Goto, K. Davidon's method for minimization problems in Hilbert space with applications to control problems. Society of Industrial Applied Mathematics Journal on Control 8: 163-178. 1970.

51. Pierson, B. L. and Rajtora, S. G. Computation experience with the Davidon method applied to optimal control problems. Engineering Research Institute preprint 65000. Iowa State University, Ames, Iowa. 1970.

52. Pierre, D. A. Optimization theory with applications. New York, New York, John Wiley and Sons, Inc. 1969.

53. Carnahan, B., Luther, H., and Wilkes, J. Applied numerical methods. New York, New York, John Wiley and Sons, Inc. 1969.

54. Kantorovich, L. V. and Akilov, G. P. Functional analysis in normed linear spaces. Translated from Russian by D. E. Brown, Edited by Dr. A. P. Robertson. New York, New York, The Macmillan Co. 1964.

55. Doerfler, T. E. The compounding of gradient error in the method of parallel tangents, M.S. thesis. Ames, Iowa, Iowa State University. 1962.

56. Stewart, G. W. A modification of Davidon's minimization method to accept difference approximations of derivatives. Journal of the Association for Computing Machinery 14: 72-83. 1967.

57. Cornick, D. E. and Seversike, L. K. Optimum parking orbit orientation for a three-dimensional capture-escape mission. Journal of Spacecraft and Rockets 7: 803-813. 1970.

58. Budak, B. M., Berkovich, E. M., and Solov'eva, E. N. Difference approximations in optimal control problems. Society of Industrial Applied Mathematics Journal on Control 7: 18-31. 1969.

59. Cullum, J. Discrete approximations to continuous optimal control problems 7: 32-49. 1969.

60. Richtmyer, R. D. and Morton, K. W. Difference methods for initial-value problems. 2nd Edition. New York, New York. Interscience Publishers. 1967.

61. Urabe, M. Convergence of numerical iteration in solution of equations. Journal of Science of the Hiroshima University Series A 19: 479-489. 1956.

62. Collatz, L. The numerical treatment of differential equations. Berlin, Germany, Springer-Verlag. 1960.

63. Michel, A. N. System mathematics. Mimeographed class notes, 595F. Ames, Iowa, Iowa State University, Department of Electrical Engineering. 1968.

64. Wilansky, A. Functional analysis. New York, New York. Blaisdell Publishing Company. 1964.

65. Hildebrand, F. B. Methods of applied mathematics. 2nd ed. Englewood Cliffs, New Jersey, Prentice-Hall, Inc. 1952.

66. Rivlin, T. J. An introduction to the approximation of functions. Waltham, Massachusetts. Balisdell Publishing Company. 1969.

67. Henrici, P. Discrete variable methods in ordinary differential equations. New York, New York, John Wiley and Sons, Inc. 1962.

ACKNOWLEDGMENTS

The author wishes to thank Dr. A. N. Michel for his enthusiastic guidance and suggestions concerning the research reported in this dissertation. In addition, he wishes to thank Dr. L. K. Seversike for his support and encouragement throughout the author's undergraduate and graduate studies, and to express his appreciation to his committee co-chairmen Dr. E. W. Anderson and Dr. C. J. Triska for their assistance during the degree program.

To his wife Mary no acknowledgment can possibly convey the appreciation of her sacrifices and understanding over the past years.

The author wishes to thank the Iowa State University Research Foundation and the Office of Naval Research for their financial support of this study, and Mrs. Gunnells for her patience throughout the typing of the manuscript.

APPENDIX A. MATHEMATICAL PRELIMINARIES

In the development of an approximation theory for the numerical optimization of distributed parameter systems, results from several areas of applied mathematics are utilized. The most important of these areas include (1) numerical analysis, (2) functional analysis, (3) optimization theory, (4) partial differential equations, and (5) approximation theory. In an attempt to make this dissertation reasonably self-contained, a limited collection of definitions and theorems from these areas will be presented. By necessity, the treatment will be brief and incomplete; only material which is used in this dissertation will be discussed. In some instantancies standard definitions and results will be altered to include concepts which are not introduced elsewhere in this appendix. It will be assumed throughout, that the reader is familiar with standard mathematical notation.

Selected Results from Functional Analysis

The natural setting for the application of gradient methods to distributed parameter systems is a real separable Hilbert space. The reasons for this are (1) gradient methods require the concept of direction (hence, an inner product) in the function space in which the iteration takes place, and (2) separability and completeness are required in the proof of convergence. Unfortunately, the definition of a Hilbert

space as "a complete inner product space" leaves much unsaid, and consequently some additional preliminary concepts need to be introduced.

Definition A-1: A linear space is a set X for which there are defined an operation of addition denoted by +, so that $\{X;+\}$ is a commutative group; and an operation of scalar multiplication satisfying the distributive law $\alpha(x+y)=\alpha x+\beta y, (\alpha+\beta)x=\alpha x+\beta x$, and $(\alpha\beta)x=\alpha(\beta x)$, $1x=x$ for all $x,y\varepsilon X$ and $\alpha,\beta\varepsilon\,\mathcal{F}$ a scalar field.

In what follows the terms linear space and vector space will be used interchangeably, and the elements of a linear (vector) space will be referred to as vectors. If $\mathcal{F}$ is the field of real numbers, then X is a real vector space; if $\mathcal{F}$ is the field of complex numbers, then X is a complex vector space.

Definition A-2: A nonempty subset S of a linear space X is called a subspace in X if x+y is in S whenever x and y are both in S and if also αx is in S whenever x is in S and α is any scalar.

Definition A-3: A functional is a mapping of a linear space X into the scalars R^1, i.e., $f: X\to R^1$.

Definition A-4: An inner product on a linear space X is a complex valued function of two elements selected from the

space X, denoted by $\langle x,y\rangle$, and satisfying the conditions:

(i) $\langle x,y\rangle$ is linear as a function of x for fixed y.

(ii) $\langle y,x\rangle = \overline{\langle x,y\rangle}$ (the complex conjugate).

(iii) $\langle x,x\rangle > 0$ if $x \neq 0$.

Definition A-5: A norm is a real function, $||\cdot||$, defined on a linear space X satisfying, for all vectors $x,y \in X$, $\alpha \in \mathbb{C}$ the conditions:

(i) $||x|| > 0 \quad x \neq 0$

(ii) $||x+y|| \leq ||x|| + ||y||$ (triangular inequality)

(iii) $||\alpha x|| = |\alpha| \, ||x||$ (homogeneity).

Definition A-6: An inner product space is a linear space together with an inner product defined on it.

Remark: In an inner product space the function $||\cdot|| = \sqrt{\langle \cdot,\cdot \rangle}$ is a norm.

Definition A-7: A Cauchy sequence in an inner product space is a sequence $\{x_n\}$ such that to each $\varepsilon > 0$, there corresponds a number N such that $||x_n - x_m|| < \varepsilon$ whenever $n > N$ and $m > N$.

Definition A-8: A normed linear space is said to be complete if every Cauchy sequence in it is convergent (a normed linear space is a linear space together with a norm).

The definition of a Hilbert space can now be given.

Definition A-9: A Hilbert space is a complete inner product space.

Definition A-10: A space is called separable if it contains a countable dense subset.

In a separable Hilbert space there exists at least one linearly independent set of vectors which spans the space. Hence every vector can be written as a countable linear combination of this linearly independent set. The partial sums of this countable linear combinations forms a Cauchy sequence; hence, converge to a unique element in the space.

Another concept which is important is that of a linear transformation.

Definition A-11: If X and Y are linear spaces, a mapping $T: X \to Y$ is a linear transformation, if for all scalars α,β, $T(\alpha y+\beta y)= \alpha Tx+\beta Ty$, for all $x,y \in X$.

Definition A-12: A linear operator is a linear transformation of X into X, i.e., $T:X \to X$.

Remark: The operator defined by

$$Lu = \int_0^{T_f}\int_0^{R_f} G_F(r,t;\xi,\tau)u(\xi,\tau)d\Omega \ ,$$

is linear (due to the properties of the definite integral).

Definition A-13: Let X and Y be normed spaces and let

$A \varepsilon B[X,Y]$. The adjoint operator $A^*: Y^* \to X^*$ is defined by the equation $\langle x, A^*y^* \rangle = \langle Ax, y^* \rangle$.

Definition A-14: An operator A defined on a Hilbert space is said to be self-adjoint if $A=A^*$.

Definition A-15: An operator A defined on a Hilbert space is said to be a projection if $A^2=A$ and $A^*=A$, where $A^2=AA$.

Definition A-16: A vector x is orthogonal to a subset M of an inner product space X, if $\langle x,y \rangle = 0$ for all $y \varepsilon M$. This is written $x \perp M$. The set of all such vectors is called the annihilator of M and is written as $M^{\perp}$. Thus

$$M^{\perp} = \{x: x \perp M, x \varepsilon X\}$$

Theorem A-1: (63) (The Projection Theorem) If M is a subspace of X, then $M+M^{\perp}=X$, where X is a Hilbert space.

Definition A-17: If S be a subset of the domain of the function f, where $x \varepsilon X$, then an evaluation map E_s is defined by $E_s f = \{f(x);\ x \varepsilon S\}$.

Definition A-18: A real valued functional f defined on a convex subset C of a linear space is said to be convex if

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2)$$

for all $x_1, x_2 \varepsilon C$ and all α, $0<\alpha<1$.

Definition A-19: $J[\cdot]$ is a quadratic functional, defined on a real Hilbert space X, if $J[\cdot]$ has the general form

$$J[x]=J_0+\langle c,x\rangle + \frac{1}{2}\langle x,Ax\rangle,$$

where A is a linear operator.

Remark: It can be shown that a quadratic functional is convex (52).

Theorem A-2: (22) (The existence of the Optimal Control) If

(i) H is a Hilbert space (e.g., L^2),

(ii) U is a closed convex bounded subset of H,

(iii) J is a real continuous convex function on U,

then there exists a $u^* \varepsilon U$ such that

$$J[u^*] = \inf_{u\varepsilon U} J[u]$$

From Theorem A-2 the solution to the problem formulated in Chapter II exists and is unique.

The following results are utilized by Chapter IV to determine the spectral bounds of an operator.

Definition A-20: The set of all complex numbers λ such that $A-\lambda I$ is invertible (has an inverse) for a given operator A, is called the resolvent set of A and is denoted by $\rho(A)$.

Definition A-21: The spectrum of an operator A is denoted by $\sigma(A)$, where $\sigma(A)=\{\lambda: \lambda \notin \rho(A)\}$.

Lemma A-1: (64) Let $A \varepsilon B[H]$, where B[H] denotes the set of bounded operators on H, a normed space, then A*A is self-adjoint.

Lemma A-2: (64) If $A \varepsilon B[H]$, then I+A*A is self-adjoint.

Lemma A-3: (63) If A is self-adjoint, then $\sigma(A) \subset R^1$.

Lemma A-4: (65) The spectrum of a skew symmetric operator is pure imaginary (A is skew symmetric if A=-A*).

Lemma A-5: Let A be a completely continuous, skew symmetric operator then

$$1 \leq \sigma(I+A^*A) .$$

Another important theorem in analysis which is especially useful in proving convergence of iteration formulas is the Fixed Point Theorem. The basis of this theorem is the concept of a contraction mapping.

Definition A-22: Let A be a mapping (not necessarily linear) of a Hilbert space H (or Banach space or even a metric space) into itself. Let for some α, $0 \leq \alpha < 1$,

$$||A(x)-A(y)|| \leq \alpha ||x-y||,$$

for all y, $y \varepsilon H$. Then A is said to be a contraction.

Theorem A-3: (63) (Fixed Point Theorem) Every contraction A, defined on a Hilbert space has one and only one fixed point, i.e., $Ax=x$ has one and only one solution for $x\varepsilon H$. (This theorem holds also for metric spaces).

In the optimal control of distributed parameter systems the extension of the concept of a derivate in a Hilbert space setting is useful.

Definition A-23: (31) Let $x\varepsilon D \subset X$ and let h be arbitrary in X. If the limit

$$\frac{dJ}{d\alpha} = \lim_{\alpha \to 0} \frac{J[x+\alpha h]-J[x]}{\alpha} ,$$

exist, it is called the Gateaux differential of J at x with increment h. If this limit exist for each $h\varepsilon X$, the functional J is said to be Gateaux differentiable at x.

Definition A-24: (31) Let J be a transformation defined on an open domain D in a normed space X and having range in a normed space Y. If for fixed $x\varepsilon D$ and $h\varepsilon X$ there exists $\delta J(x;h)\varepsilon Y$ which is linear and continuous with respect to h such that

$$\lim_{||h|| \to 0} \frac{||J(x+h)-J(x)-\delta J(x;h)||}{||h||} = 0,$$

then J is said to be Frećhet differentiable at x and $\delta J(x;h)$ is said to be the Frećhet differential of J at x with increment h.

Pertinent Results from Partial Differential Equations

The concept of well-posedness plays an important role in the treatment of partial differential equations. It is analogous to the concepts of existence and uniqueness of solutions in ordinary differential equations. In this dissertation an initial-value problem will be called well-posed if

(i) a solution exist,

(ii) the solution is unique,

(iii) the solution depends continuously on the initial data.

Reference (60) contains a detailed discussion of well-posedness of the abstract Cauchy initial value problem.

Another important result used in Chapter II is the representation of the solution of a partial differential equation in terms of linear operators

$$x(r,t)=\Phi(t)x_0 + S^{-1}(t)u_d + T^{-1}(t)u_b \ . \qquad \text{(A-1)}$$

Balakrishnan (20) has given conditions under which Equation A-1 represents the solution to the nonhomogenous partial differential equation. In this work only problems for which the Green's function yields the representation given in Equation A-1 are considered.

For example, consider the one-dimensional, non-homogenous wave equation defined by

$$x_{tt} = x_{rr} + u_d, \tag{A-2}$$

$$x(r,0)=0, \tag{A-3}$$

$$x_t(r,0)=0, \tag{A-4}$$

$$x(0,t)=0, \tag{A-5}$$

$$x(1,t)=0, \tag{A-6}$$

This second order system can be rewritten as an equivalent first order system by considering the transformation defined by $v = \frac{\partial x}{\partial t}$ and $w = \frac{\partial x}{\partial r}$; hence, the above system becomes

$$\begin{bmatrix} \frac{\partial w}{\partial t} \\ \frac{\partial v}{\partial t} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\partial}{\partial r} \\ \frac{\partial}{\partial r} & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ u_d \end{bmatrix} . \tag{A-7}$$

In order to obtain the Green's function for this problem, the following eigenvalue problem is considered

$$\begin{bmatrix} 0 & \frac{\partial}{\partial r} \\ \frac{\partial}{\partial r} & 0 \end{bmatrix} \begin{bmatrix} \phi_k^1(r) \\ \phi_k^2(r) \end{bmatrix} = \lambda_k \begin{bmatrix} \phi_k^1(r) \\ \phi_k^2(r) \end{bmatrix} , \tag{A-8}$$

where $\phi_k^2(0) = \phi_k^2(1)=0$. The solution of this eigenvalue problem yields $\lambda_k = \pm k\pi i$ as the eigenvalues, and

$$\begin{bmatrix} \phi_k^1(r) \\ \phi_k^2(r) \end{bmatrix} = \begin{bmatrix} -i \cos(k\pi r) \\ \sin(k\pi r) \end{bmatrix} \tag{A-9}$$

as the eigenfunctions. By using these eigenfunctions, the Green's function for this problem is given by

$$G_F(r,t;\xi,\tau)=$$

$$2\begin{bmatrix} \sum_{k=1}^{\infty} \cos k\pi(t-\tau)\cos k\pi r\cos k\pi\xi & \sum_{k=1}^{\infty} \sin k\pi(t-\tau)\cos k\pi r\sin k\pi\xi \\ -\sum_{k=1}^{\infty} \sin k\pi(t-\tau)\sin k\pi r\cos k\pi\xi & \sum_{k=1}^{\infty} \cos k\pi(t-\tau)\sin k\pi r\sin k\pi\xi \end{bmatrix} \quad \text{(A-10)}$$

which is represented by

$$G_F(r,t;\xi,\tau)=\begin{bmatrix} G_{11}(r,t;\xi,\tau) & G_{12}(r,t;\xi,\tau) \\ G_{21}(r,t;\xi,\tau) & G_{22}(r,t;\xi,\tau) \end{bmatrix} . \quad \text{(A-11)}$$

The solution of Equation A-7 can then be written as

$$\begin{bmatrix} w(r,t) \\ v(r,t) \end{bmatrix} = \int_0^t \int_0^1 \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} 0 \\ u_d(\xi,\tau) \end{bmatrix} d\xi d\tau . \quad \text{(A-12)}$$

From the above relation the Green's function for the original problem can be determined as follows:

$$x(r,t) = \int_0^r \frac{\partial x}{\partial r}\, dr = \int_0^r w(r,t)dr \qquad \text{(A-13)}$$

$$= \int_0^r\int_0^t\int_0^1 G_{12}\, u_d(\xi,\tau)d\xi d\tau dr$$

$$= \int_0^t\int_0^1\int_0^r G_{i2}\, u_d\, drd\xi d\tau$$

$$= \int_0^t\int_0^1 [\int_0^r \sum_{k=1}^{\infty} 2\sin k\pi(t-\tau)\sin k\pi\xi\, \cos k\pi r dr] u_d(\xi,\tau)d\xi d\tau,$$

by performing the above indicate integration, the Green's function is determined as

$$G_F^*(r,t;\xi,\tau) = \sum_{k=1}^{\infty} \frac{2}{k\pi} \sin k\pi(t-\tau)\sin k\pi\xi\, \sin k\pi r. \qquad \text{(A-14)}$$

Numerical investigation of the convergence of this series indicates that more than ten terms of this series are required to obtain three significant digits of accuracy.

Pertinent Results from Approximation Theory and Numerical Analysis

The fundamental problem of approximation theory may loosely be stated as follows: determine $\tilde{x}^* \varepsilon \tilde{X}$, where $\tilde{X}$ is a subspace of a linear space X, such that its distance from $x^* \varepsilon X$ is a minimum, that is, find $\tilde{x}^*$ such that $||x^*-\tilde{x}||$ is minimized. The following theorem is the basis of the approximation theory developed in Chapter IV.

Theorem A-4: (66) If X is a normed linear space and $\tilde{X}$ is a finite-dimensional subspace of X, then given $x^* \varepsilon X$, there exists $\tilde{x}^* \varepsilon \tilde{X}$ such that

$$||x^*-\tilde{x}^*|| \leq ||x^*-\tilde{x}|| \quad \text{for all } \tilde{x}\varepsilon\tilde{X}.$$

The following results are due to Kantorovich (54). Let $\tilde{X}$ be a complete subspace of the Hilbert space X, and let P denote a projection from X onto $\tilde{X}$, i.e., $PX=\tilde{X}$, $P^2=P$, then clearly P does not alter the elements of $\tilde{X}$. Consider two iterations, the first in the space X

$$x_{n+1}=G(x_n), \tag{A-15}$$

and the second in the space $\tilde{X}$

$$\tilde{x}_{n+1} = \tilde{G}(\tilde{x}_n). \tag{A-16}$$

In what follows Equation A-15 will be called the exact iteration, and Equation A-16 the approximate iteration.

The spaces X and $\tilde{X}$, and the functions G and $\tilde{G}$ will be connected by the following conditions.

(i) (Condition that G and $\tilde{G}$ be neighboring functions) for every $\tilde{x}\varepsilon\tilde{X}$,

$$||PG(\tilde{x}) - \tilde{G}(\tilde{x})|| \leq \eta ||\tilde{x}|| . \tag{A-17}$$

(ii) (Condition for elements of the form G(x) to be approximated closely by elements of $\tilde{X}$.) For every $x\varepsilon X$, there is an $\tilde{x}\varepsilon\tilde{X}$ such that

$$||G(x)-\tilde{x}|| \leq \eta_1 ||x||.$$

The above results are useful in answering problems encountered in this dissertation. These problems include: (1) establishing the practicality and convergence of the approximated gradient algorithms; (2) investigating the speed of convergence; and, (3) obtaining suitable estimates of the error.

In the analysis of the approximations due to discretization, the approximate operator $\tilde{G}$ often depends on a parameter h, which is a measure of the discretization. By extending the definition due to Henrici (67), the order of an approximate operator can be defined as follows; if $\tilde{G}$ is of <u>order p</u> then

$$\tilde{G}[x;h] = c_{p+1} h^{p+1} x^{(p+1)}(x+th) + 0(h^{p+2}) \quad .$$

where c_{p+1} is a constant, $0 \leq t \leq 1$, and $\lim_{h \to 0} \frac{0(h)}{h} = 0$.

This definition is standard in connection with the approximation of differential operators by difference operators, and integral operators by summation operators.

The approximation of partial differential equations by finite-difference operators results in errors. Before discussing these errors some additional nomenclature is required.

Definition A-25: Let $\Omega x T$ denote the domain of a finite-difference operator. Then the net (mesh, grid) which partitions $\Omega x T$ is defined by

$$\mathcal{N} = \{(r,t);\ r=a_i \text{ and } t = b_i\}.$$

The nodes N of the net $\mathcal{N}$ are the points of intersection of the curves which define the net (see Figure A-1).

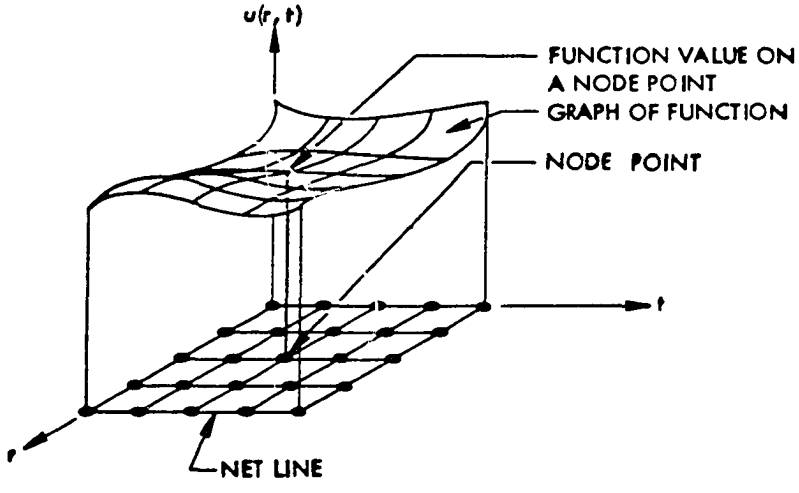


Figure A-1. The node N of the net $\mathcal{N}$

Definition A-26: Let x(ir,it) denote the exact solution of a partial differential equation evaluated at the nodes of the net. Then the truncation error (on the nodes) of the finite-difference operator is defined as

$$e_t(ir,it) = x(ir,it) - \tilde{x}(ir,it),$$

where $\tilde{x}$ is the approximate solution obtained from the finite-difference method.

The second source of errors arises from the fact that $\tilde{x}$ cannot be calculated with exact precision because of the limited accuracy of any computing equipment.

Let $\hat{x}(ir,it)$ denote the values that are actually calculated by the computer. The difference

$$e_r(ir,it)=\hat{x}(ir,it)-\tilde{x}(ir,it),$$

is called the round-off error.

In addition to truncation errors, and round-off errors there are interpolation errors. Interpolation errors are the errors due to the approximation of functions (e.g., surfaces, etc.) by interpolating functions. Let P denote the projection from some class of functions, which form a Hilbert space, to the class of approximating functions. Then $||x-Px||$ is a measure of the accuracy of the interpolating functions.

APPENDIX B. THE NON-LINEAR DISTRIBUTED PARAMETER OPTIMAL CONTROL PROBLEM

Problem Formulation

Consider the nonlinear distributed dynamical system

$$\frac{\partial x(r,t)}{\partial t} = f(t,r,x(r,t), \frac{\partial x}{\partial r}, \ldots, \frac{\partial^k x}{\partial r^k}, u_D(r,t)), \qquad \text{(B-1)}$$

with initial conditions $x(r,t_0)=x_0(r)$, and with boundary condition

$$\frac{\partial x(r,t)}{\partial t}\Big|_{\partial\Omega} = g(t,r,x, \frac{\partial x}{\partial r}, \ldots, \frac{\partial^{k-1}}{\partial r^{k-1}}, u_B(t))\Big|_{\partial\Omega}, \qquad \text{(B-2)}$$

where $r\varepsilon\Omega\subset R^m$, $t\varepsilon T\subset R^1$.

At any time $t\varepsilon T$, the state of the system is denoted by $x(r,t)$, the distributed control vector is denoted by $u_D(r,t)$, and the boundary control vector is represented by $u_B(t)$. The i^{th} component of $x(r,t)$ is written as $x_i(r_1,r_2,\ldots,r_m,t)\varepsilon L^2(\Omega xT)$, $i=1,2,\ldots,n$; the k^{th} component of $u_D(r,t)$ is denoted by $u_D^i(r_1,r_2,\ldots,r_m,t)\varepsilon L^2(\Omega xT)$, $i=1,2,\ldots,p\leq n$; and the i^{th} component of $u_B(t)$ is represented as $u_B^i(t)\varepsilon L^2(T)$, $i=1,2,\ldots,k\leq n$. If $\partial\Omega$ denotes the boundary of Ω, then $r_b\varepsilon\partial\Omega$.

The notation $\partial^k x(r,t)/\partial r^k$ is utilized to represent spatial derivatives of $x(r,t)$ (refer to (28)).

Only well-posed distributed parameter systems with unique solutions are considered.

The performance functional considered is given by

$$J[u_D(r,t),u_B(t)] = G[x,r,T_f)]\Big|_{\partial\Omega} + \overline{\int}_{\Omega} K[x(r,T_f)]d\Omega$$

$$+ \int_T L_B(t,r,x, \frac{\partial x}{\partial r}, \ldots, \frac{\partial^k x}{\partial r^k}, u_B)\Big|_{\partial\Omega} dt \qquad \text{(B-3)}$$

$$+ \int_T \overline{\int}_{\Omega} L_D(t,r,x, \frac{\partial x}{\partial r}, \ldots, \frac{\partial^k x}{\partial r^k}, u_D)d\Omega\, dt,$$

where

$$\overline{\int}_{\Omega}(\,)d\Omega \equiv \int_{r_1}\int_{r_2}\cdots\int_{r_m}(\,)\, dr_1 dr_2 \cdots dr_m,$$

and L_D, K, and G are sufficiently smooth real valued functions.

The optimum control problem is now stated as follows: determine from the set U of admissible controls the control vector u,

$$u^T = [u_D^T, u_B^T],\ u\varepsilon U, \qquad \text{(B-4)}$$

which satisfies the state equations along with the initial and the boundary conditions and at the same time minimizes $J[u_D,u_B]$.

Derivation of the Necessary Conditions

Let J be the functional defined by Equation B-3. In order for J to have a relative local minimum at $u^*\varepsilon U$, it is necessary that the first Frećhet derivative of J at u^* be identically zero, that is

$$\left.\frac{\partial J}{\partial u}\right|_{u^*} = \lim_{u \to u^*} \frac{J[u]-J[u^*]}{||u-u^*||} = 0 \ . \qquad \text{(B-5)}$$

Let $p(r,t)$ denote the Lagrange multiplier associated with the dynamical system, and let $\lambda(t)$ denote the Lagrange multiplier associated with the boundary conditions. By using the Lagrange multipliers $p(r,t)$ and $\lambda(t)$, the constrained problem defined by Equations B-1, B-2, and B-3 can be reformulated as an unconstrained problem, where the unconstrained cost functional is given by

$$\hat{J}[u]=G + \int_T \{L_B + \langle \lambda, g - \frac{\partial x}{\partial t} \rangle \Big|_{\partial\Omega} \} dt + \int_{\Omega} K \, d\Omega \qquad \text{(B-6)}$$

$$+ \int_T \int_{\Omega} \{L_D + \langle p, f - \frac{\partial x}{\partial t} \rangle d\Omega \, dt.^1$$

If the constraints are satisfied then $\min_{u \in U} \hat{J} = \min_{u \in U} J$. Thus, the minimum of J can be obtained from the minimum of $\hat{J}$. In the following development it will be assumed that the constraints are satisfied and hence, $\hat{J}=J$.

Let H_D denote the distributed Hamiltonian, and let H_B denote the boundary Hamiltonian, where

$$H_D = L_D + \langle p, f \rangle, \qquad \text{(B-7)}$$

and

$$H_B = L_B + \langle \lambda, g \rangle. \qquad \text{(B-8)}$$

---

[1] The arguments of G, L_B, λ, g, f, K, L_D, p, and x will be dropped for notational simplicity.

Substitution of the distributed and the boundary Hamiltonians into Equation B-6 yields

$$J[u]=G+\int_{\Omega} K\, d\Omega + \int_{T}[H_B-\langle\lambda, \frac{\partial x}{\partial t}\Big|_{\partial\Omega}\rangle]dt$$

$$+ \int_{T}\int_{\Omega} [H_D-\langle p, \frac{\partial x}{\partial t}\rangle]d\Omega\, dt\ . \qquad \text{(B-9)}$$

From Equation B-8 the difference $J[u]-J[u^*]$ is given by

$$J[u]-J[u^*]=G-G^*+\int_{\Omega}(K-K^*)d\Omega$$

$$+ \int_{T}[H_B-H_B^*-\langle\lambda, \frac{\partial x}{\partial t}\Big|_{\partial\Omega} - \frac{\partial x^*}{\partial t}\Big|_{\partial\Omega}\rangle]dt$$

$$+ \int_{T}\int_{\Omega}[H_D-H_D^*-\langle p,\frac{\partial x}{\partial t} - \frac{\partial x^*}{\partial t}\rangle]d\Omega\, dt. \qquad \text{(B-10)}$$

Now perturb the control u^* by letting $u=u^*+\varepsilon\mu$, and let $x=x^*+\varepsilon\Psi$ denote the trajectory corresponding to u. Expanding the terms in Equation B-9 about u^* and x^* in a Taylor's series yields

$$G-G^*=\varepsilon\langle\frac{\partial G}{\partial x}\Big|_*, \Psi\rangle\Big|_{\partial\Omega}\Big|_{T_f} + \ldots + 0(\varepsilon), \qquad \text{(B-11)}$$

$$K-K^*=\varepsilon\langle\frac{\partial K}{\partial x}\Big|_*, \Psi\rangle\Big|_{T_f} + \ldots + 0(\varepsilon), \qquad \text{(B-12)}$$

$$H_B - H_B^* = \varepsilon \left\langle \frac{\partial H_B}{\partial x}\Big|_*, \Psi \right\rangle \Big|_{\partial\Omega} + \varepsilon \sum_{i=1}^{k-1} \left\langle \frac{\partial H_B}{\partial x_{r^i}}, \Psi_{r^i} \right\rangle \Big|_{\partial\Omega} + \varepsilon \left\langle \frac{\partial H_B}{\partial u_B}, \beta \right\rangle$$

$$+ 0(\varepsilon) \tag{B-13}$$

$$H_D - H_D^* = \varepsilon \left\langle \frac{\partial H_D}{\partial x}\Big|_*, \Psi \right\rangle + \varepsilon \sum_{i=1}^{k} \left\langle \frac{\partial H_D}{\partial x_{r^i}}\Big|_*, \frac{\partial^k \psi}{\partial r^i} \right\rangle + \varepsilon \left\langle \frac{\partial H_D}{\partial u_D}, \eta \right\rangle + 0(\varepsilon), \tag{B-14}$$

and

$$\frac{\partial x}{\partial t} - \frac{\partial x^*}{\partial t} = \varepsilon \frac{\partial \Psi}{\partial t}, \tag{B-15}$$

where

$$\mu^T = [\eta^T, \beta^T], \text{ and } \lim_{\varepsilon \to 0} \frac{0(\varepsilon)}{\varepsilon} = 0.$$

Substitution of Equations B-11, B-12, B-13, B-14, and B-15 into B-10, and integrating by parts yields

$$\lim_{\varepsilon \to 0} \frac{J[u] - J[u^*]}{\varepsilon} = \left\langle \frac{\partial G}{\partial x} - \lambda, \Psi \right\rangle \Big|_{\partial\Omega} \Big|_{T_f} + \int_\Omega \left\langle \frac{\partial K}{\partial x} - p, \Psi \right\rangle \Big|_{T_f} d\Omega \tag{B-16}$$

$$+ \int_T \{ \left\langle \sum_{i=0}^{k-1} (-1)^i \frac{\partial^i}{\partial r^i} \left[\frac{\partial H_D}{\partial x_{r^{i+1}}} \right] + \dot{\lambda} + \frac{\partial H_B}{\partial x}, \Psi \right\rangle \Big|_{\partial\Omega} + \left\langle \frac{\partial H_D}{\partial x_{r^k}}, \Psi_{r^{k-1}} \right\rangle \Big|_{\partial\Omega}$$

$$+ \sum_{i=2}^{k-1} \left\langle \sum_{j=0}^{k=i} (-1)^j \frac{\partial^j}{\partial r^i} \left[\frac{\partial H_D}{\partial x_{r^{i+j}}} \right] + \frac{\partial H_B}{\partial x_{r^{i-1}}}, \Psi_{r^{i-1}} \right\rangle \Big|_{\partial\Omega} \} dt$$

$$+ \int_T \int_\Omega \left[\left\langle \frac{\partial H_D}{\partial x} + \sum_{i=1}^{k} (-1)^i \frac{\partial^i}{\partial r^i} \left[\frac{\partial H_D}{\partial x_{r^i}} \right] + \frac{\partial p}{\partial t}, \Psi \right\rangle + \left\langle \frac{\partial H_D}{\partial u_b}, \eta \right\rangle \right] d\Omega \, dt.$$

Optimality Conditions

It follows from Equation B-15 that in order for $[u_0^{*T}, u_0^{*T}]$ to be optimal, it is necessary that there exist functions $p^*(r,t)$ and $\lambda^*(t)$ such that

a. $x^*(r,t)$ is a solution of the distributed state system

$$\frac{\partial x}{\partial t} = f,\ x(r,t_0) = x_0(r),\ \frac{\partial x}{\partial t}\Big|_{\partial\Omega} = g\Big|_{\partial\Omega}. \tag{B-17}$$

b. $p^*(r,t)$ is a solution of the distributed costate system

$$\frac{\partial p}{\partial t} = -\left[\frac{\partial H_D}{\partial x} + \sum_{i=1}^{k} (-1)^i \frac{\partial^i}{\partial r^i}\left[\frac{\partial H_D}{\partial x_{r^i}}\right]\right],\ p(r,T_f) =$$

$$\frac{\partial K}{\partial x(r,t)}\Big|_{t=T_f}. \tag{B-18}$$

c. $\lambda^*(t)$ is a solution of the ordinary differential equation

$$\frac{d\lambda}{dt} = -\left[\frac{\partial H_B}{\partial x} + \sum_{i=0}^{k-1} (-1)^i \frac{\partial^i}{\partial r^i}\left[\frac{\partial H_D}{\partial x_{r^{i+1}}}\right]\right]\Bigg|_{\partial\Omega}. \tag{B-19}$$

$$\lambda(T_f) = \frac{\partial G}{\partial x}\Big|_{\partial\Omega}\Big|_{t=T_f}$$

d. The transversality conditions are satisfied, i.e.,

$$\left.\frac{\partial H_D}{\partial x_{r^k}}\right|_{\partial\Omega}=0,\ \left.\frac{\partial H_B}{\partial x_{r^{i-1}}}\right|_{\partial\Omega}+\sum_{j=0}^{k-i}(-1)^j\frac{\partial^j}{\partial r^j}\left.\frac{\partial H_D}{\partial x_{r^{i+j}}}\right|_{\partial\Omega}=0 \tag{B-20}$$

$i=2,3,\ldots,k-1.$

e. The gradient of J vanishes, i.e.,

$$g(u)^T=[g_D(u)^T,g_B(u)^T]=[(\partial H_D/\partial u_D)^T, \tag{B-21}$$

$$(\partial H_B/\partial u_B)^T]=[0^T,0^T].$$